

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日      2002年10月  8日  
Date of Application:

出願番号      特願2002-295378  
Application Number:

[ST. 10/C] :      [JP2002-295378]

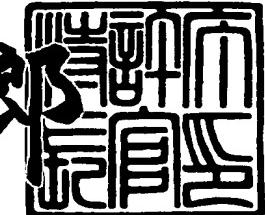
出願人      株式会社リコー  
Applicant(s):

87

2003年 7月 8日

特許庁長官  
Commissioner,  
Japan Patent Office

太田信一



【書類名】 特許願  
【整理番号】 0207132  
【提出日】 平成14年10月 8日  
【あて先】 特許庁長官 太田 信一郎 殿  
【国際特許分類】 G03G 21/00 370  
【発明の名称】 画像形成装置および操作表示部プログラム生成方法  
【請求項の数】 24  
【発明者】  
【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内  
【氏名】 大石 勉  
【発明者】  
【住所又は居所】 東京都大田区中馬込1丁目3番6号 株式会社リコー内  
【氏名】 杉浦 裕子  
【特許出願人】  
【識別番号】 000006747  
【氏名又は名称】 株式会社リコー  
【代理人】  
【識別番号】 100070150  
【弁理士】  
【氏名又は名称】 伊東 忠彦  
【先の出願に基づく優先権主張】  
【出願番号】 特願2002-218814  
【出願日】 平成14年 7月26日  
【手数料の表示】  
【予納台帳番号】 002989  
【納付金額】 21,000円  
【提出物件の目録】  
【物件名】 明細書 1  
【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9911477

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 画像形成装置および操作表示部プログラム生成方法

【特許請求の範囲】

【請求項 1】 画面表示および操作入力を行う操作表示部を有するオペレーショングループを備えた画像形成装置であって、

利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させるプログラム入力手段と、

前記プログラム入力手段によって入力されたカスタマイズプログラムをステップごとに逐次実行するプログラム実行手段と、

を備えたことを特徴とする画像形成装置。

【請求項 2】 前記カスタマイズプログラムを記憶するための予め定められたカスタマイズ領域を有する記憶手段をさらに備え、

前記プログラム入力手段は、さらに前記入力画面から入力されたカスタマイズプログラムを前記カスタマイズ領域に格納し、

前記プログラム実行手段は、前記カスタマイズ領域に格納された前記カスタマイズプログラムをステップごとに逐次実行することを特徴とする請求項 1 に記載の画像形成装置。

【請求項 3】 前記プログラム入力手段は、さらに前記操作表示部に表示されるとともに操作入力を行うキーとカスタマイズプログラムとの対応付けを行うことを特徴とする請求項 1 または 2 に記載の画像形成装置。

【請求項 4】 前記プログラム入力手段は、さらに入力されたカスタマイズプログラムを動作確認のために実行させることを特徴とする請求項 1 ~ 3 のいずれか一つに記載の画像形成装置。

【請求項 5】 前記プログラム入力手段は、前記表示手段としてのネットワークに接続されたクライアント端末に対し前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする請求項 1 ~ 4 のいずれか一つに記載の画像形成装置。

【請求項 6】前記プログラム入力手段は、前記表示手段としての前記オペレーションパネルの操作表示部に前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする請求項 1～4 のいずれか一つに記載の画像形成装置。

【請求項 7】画像形成装置の起動時にアプリケーションプログラムを起動する起動手段をさらに備え、

前記起動手段は、前記プログラム実行手段を起動することを特徴とする請求項 1～6 のいずれか一つに記載の画像形成装置。

【請求項 8】前記プログラム実行手段は、汎用 OS の一部として動作するインタプリタであることを特徴とする請求項 1～7 のいずれか一つに記載の画像形成装置。

【請求項 9】前記プログラム実行手段は、汎用 OS の配下で動作するインタプリタであることを特徴とする請求項 1～7 のいずれか一つに記載の画像形成装置。

【請求項 10】前記画像形成装置は、画像形成処理で使用されるハードウェア資源と、画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと前記ハードウェア資源との間に介在するコントロールサービスとを更に有し、

前記プログラム実行手段及び前記カスタマイズプログラムは、前記アプリケーションとして動作することを特徴とする請求項 1～7 のいずれか一つに記載の画像形成装置。

【請求項 11】前記カスタマイズプログラムは、前記プログラム実行手段により、前記コントロールサービスに対する関数呼び出しを行うことを特徴とする請求項 10 に記載の画像形成装置。

【請求項 12】画面表示および操作入を行なう操作表示部を有するオペレーションパネルを備えた画像形成装置であって、

利用者に対する画面表示および入力を行なわせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示し、前記入力画面から前記カスタマイズプログラムを

入力させるプログラム入力手段と、

前記カスタマイズプログラムを、前記操作表示部に表示されるとともに入力を  
行うキーに割り付ける割付手段と、

前記操作表示部から前記割付手段によって割り付けられたキーの操作入力があ  
った場合に、前記プログラム入力手段によって入力されたカスタマイズプログラ  
ムをステップごとに逐次実行するプログラム実行手段と、

を備えたことを特徴とする画像形成装置。

**【請求項13】** 前記割付手段は、画像形成装置の起動時に前記カスタマイ  
ズプログラムを、前記キーに割り付けることを特徴とする請求項12に記載の画  
像形成装置。

**【請求項14】** 前記割付手段は、前記キーと前記カスタマイズプログラム  
の割り付けを設定するキー割付け画面を表示し、前記キー割付け画面からの設定  
内容に基づいて前記カスタマイズプログラムを前記キーに割り付けることを特徴  
とする請求項13に記載の画像形成装置。

**【請求項15】** 画像形成装置における画面表示および操作入力を行うオペ  
レーションパネルの操作表示部をカスタマイズする操作表示部プログラム生成方  
法であって、

利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示  
部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプ  
ログラムの入力画面を表示し、前記入力成画面から前記カスタマイズプログラを  
入力させるプログラム入力ステップと、

前記プログラム入力ステップによって入力されたカスタマイズプログラムをス  
テップごとに逐次実行するプログラム実行ステップと、  
を含むことを特徴とする操作表示部プログラム生成方法。

**【請求項16】** 前記プログラム入力ステップは、さらに前記入力画面から  
入力されたカスタマイズプログラムを、記憶手段の予め定められたカスタマイズ  
領域に格納し、

前記プログラム実行ステップは、前記カスタマイズ領域に格納された前記カス  
タマイズプログラムをステップごとに逐次実行することを特徴とする請求項15

に記載の操作表示部プログラム生成方法。

【請求項17】 前記プログラム入力ステップは、さらに前記操作表示部に表示されるとともに操作入力を行うキーとカスタマイズプログラムとの対応付けを行うことを特徴とする請求項15または16に記載の操作表示部プログラム生成方法。

【請求項18】 前記プログラム入力ステップは、さらに入力されたカスタマイズプログラムを動作確認のために実行させることを特徴とする請求項15～17のいずれか一つに記載の操作表示部プログラム生成方法。

【請求項19】 前記プログラム入力ステップは、前記表示手段としてのネットワークに接続されたクライアント端末に対し前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする請求項15～18のいずれか一つに記載の操作表示部プログラム生成方法。

【請求項20】 前記プログラム入力ステップは、前記表示手段としての前記オペレーションパネルの操作表示部に前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする請求項15～18のいずれか一つに記載の操作表示部プログラム生成方法。

【請求項21】 画像形成装置の起動時に、画像形成装置が有するアプリケーションプログラムを起動する起動ステップをさらに含み、

前記起動ステップは、前記プログラム実行ステップを実行するプログラムを起動することを特徴とする請求項15～20のいずれか一つに記載の操作表示部プログラム生成方法。

【請求項22】 画像形成装置における画面表示および操作入力を行うオペレーションパネルの操作表示部をカスタマイズする操作表示部プログラム生成方法であって、

利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させるプログラム入力ステップと、

前記カスタマイズプログラムを、前記操作表示部に表示されるとともに入力を

行うキーに割り付ける割付ステップと、

前記操作表示部から前記割付ステップによって割り付けられたキーの操作入力があった場合に、前記プログラム入力ステップによって入力されたカスタマイズプログラムをステップごとに逐次実行するプログラム実行ステップと、

を含むことを特徴とする操作表示部プログラム生成方法。

【請求項 2 3】 前記割付ステップは、画像形成装置の起動時に前記カスタマイズプログラムを、前記キーに割り付けることを特徴とする請求項 2 2 に記載の操作表示部プログラム生成方法。

【請求項 2 4】 前記割付ステップは、前記キーと前記カスタマイズプログラムの割り付けを設定するキー割付け画面を表示し、前記キー割付け画面からの設定内容に基づいて前記カスタマイズプログラムを前記キーに割り付けることを特徴とする請求項 2 3 に記載の操作表示部プログラム生成方法。

#### 【発明の詳細な説明】

##### 【0001】

##### 【発明の属する技術分野】

この発明は、コピー、プリンタ、スキャナおよびファクシミリなどの画像形成処理にかかるユーザサービスを提供し、操作表示部のカスタマイズを行うことができる画像形成装置および操作表示部プログラム生成方法に関するものである。

##### 【従来の技術】

近年では、プリンタ、コピー、ファクシミリ、スキャナなどの各装置の機能を1つの筐体内に収納した画像形成装置（以下、「複合機」という。）が知られている。この複合機は、1つの筐体内に表示部、印刷部および撮像部などを設けるとともに、プリンタ、コピーおよびファクシミリ装置にそれぞれ対応した3種類のソフトウェアを設け、これらのソフトウェアを切り替えることによって、当該装置をプリンタ、コピー、スキャナまたはファクシミリ装置として動作させるものである。

##### 【0002】

このような従来の複合機では、利用者に対し複合機の各種操作を行わせるためのオペレーションパネルが設けられ、このオペレーションパネルに操作画面の表

示およびタッチ操作入力を行わせるための操作表示部を表示している。従来の複合機では、予め提供される機能が定められていたため、利用者が複合機に対して行う操作も大きな変更はなく、操作表示部の画面表示やタッチ操作による動作をカスタマイズする必要性はない。このため、従来の複合機には操作表示部のカスタマイズ機能は搭載されていなかった。

#### 【特許文献 1】

特開 2002-152446 号公報

#### 【0003】

##### 【発明が解決しようとする課題】

ところで、このような従来の複合機では、プリンタ、コピー、スキャナおよびファクシミリ装置に対応するソフトウェアをそれぞれ別個に設けているため、各ソフトウェアの開発に多大の時間を要する。このため、出願人は、表示部、印刷部および撮像部などの画像形成処理で使用されるハードウェア資源を有し、プリンタ、コピーまたはファクシミリなどの各ユーザサービスにそれぞれ固有の処理を行うアプリケーションを複数搭載し、これらのアプリケーションとハードウェア資源との間に介在して、ユーザサービスを提供する際に、アプリケーションの少なくとも 2 つが共通的に必要とするハードウェア資源の管理、実行制御並びに画像形成処理を行う各種コントロールサービスからなるプラットホームを備えた画像形成装置（複合機）を発明した。

#### 【0004】

このような新規な複合機では、アプリケーションと、ハードウェア資源にアクセスするような開発が難しい処理を行うコントロールサービスとを別個に設けているため、複合機の出荷後にユーザもしくは第三者であるサードベンダが画像形成処理などにかかるアプリケーションとして新規な外部アプリケーションを開発して複合機に搭載可能な構成となっている。

#### 【0005】

このため、新たに開発された外部アプリケーションの起動や実行中に操作表示部における画面表示やタッチ操作によって、予め提供されている機能とは異なる実行処理を行わせたい場合がある。このように、新規な複合機では、操作表示部

のカスタマイズという、出荷後に外部アプリケーションを搭載することを想定していない従来の複合機では問題にならなかった新規な課題が生じてくる。

#### 【0006】

このような操作表示部のカスタマイズを、外部アプリケーションの開発段階でソースコードを記述し、ソースコードをコンパイルおよびリンクすることによって行うことが考えられる。すなわち、外部アプリケーションの開発段階で、外部アプリケーション自体の処理を開発しながら操作表示部のタッチ操作に伴う実行処理を開発する場合が多くなる。この場合、操作表示部のタッチ操作に伴う実行処理の検証を逐次行いながら外部アプリケーション自体の開発を進めていくことがプログラム開発効率上好ましい。しかしながら、タッチ操作に伴う実行処理に障害を検出するたびに、ソースコードの修正、再コンパイル、再リンクを行って、開発対象の複合機に再インストールし、動作検証を行わなければならないとすると、プログラム開発の効率が悪くなるという問題がある。

#### 【0007】

この発明は上記に鑑みてなされたもので、オペレーションパネルの操作表示部のカスタマイズを容易に行い、プログラム開発効率を向上させることができる画像形成装置および操作表示部プログラム生成方法を得ることを目的とする。

#### 【課題を解決するための手段】

上記目的を達成するため、請求項1にかかる発明は、画面表示および操作入力を行う操作表示部を有するオペレーションパネルを備えた画像形成装置であって、利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示し、前記入力成画面から前記カスタマイズプログラムを入力させるプログラム入力手段と、前記プログラム入力手段によって入力されたカスタマイズプログラムをステップごとに逐次実行するプログラム実行手段と、を備えたことを特徴とする。

#### 【0008】

この請求項1にかかる発明によれば、プログラム入力手段によって、利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入

力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示して、この入力画面からカスタマイズプログラムを入力させ、プログラム実行手段によって、入力されたカスタマイズプログラムをステップごとに逐次実行することで、操作表示部のカスタマイズプログラムの開発において、表示手段の入力画面から追加、修正ができる、また追加修正による再コンパイル、再リンクなしに画像形成装置で実行でき、操作表示部のカスタマイズを容易に行うことができる。

#### 【0009】

また、請求項2にかかる発明は、請求項1に記載の画像形成装置において、前記カスタマイズプログラムを記憶するための予め定められたカスタマイズ領域を有する記憶手段をさらに備え、前記プログラム入力手段は、さらに前記入力画面から入力されたカスタマイズプログラムを前記カスタマイズ領域に格納し、前記プログラム実行手段は、前記カスタマイズ領域に格納された前記カスタマイズプログラムをステップごとに逐次実行することを特徴とする。

#### 【0010】

この請求項2にかかる発明によれば、プログラム入力手段によって、入力画面から入力されたカスタマイズプログラムをカスタマイズ領域に格納し、プログラム実行手段によって、カスタマイズ領域に格納されたカスタマイズプログラムをステップごとに逐次実行することで、予めカスタマイズ領域を定め当該カスタマイズ領域にカスタマイズプログラムを格納しておけば、プログラム実行時に実行するカスタマイズプログラムを容易に特定することができ、操作表示部のカスタマイズおよびその検証を容易に行うことができる。

#### 【0011】

また、請求項3にかかる発明は、請求項1または2に記載の画像形成装置において、前記プログラム入力手段は、さらに前記操作表示部に表示されるとともに操作入力を行うキーとカスタマイズプログラムとの対応付けを行うことを特徴とする。

#### 【0012】

この請求項3にかかる発明によれば、プログラム入力手段によって、操作表示

部に表示され操作入力を行うキーとカスタマイズプログラムとの対応付けを行ことで、キー操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができる。

#### 【0013】

また、請求項4にかかる発明は、請求項1～3のいずれか一つに記載の画像形成装置において、前記プログラム入力手段は、さらに入力されたカスタマイズプログラムを動作確認のために実行させることを特徴とする。

#### 【0014】

この請求項4にかかる発明によれば、プログラム入力手段によって、入力されたカスタマイズプログラムを動作確認のために実行させることで、カスタマイズプログラムを実際に画像形成装置で動作させる前に、表示手段上で簡易な検証を行うことができ、操作表示部のカスタマイズを効率的に行うことができる。

#### 【0015】

また、請求項5にかかる発明は、請求項1～4のいずれか一つに記載の画像形成装置において、前記プログラム入力手段は、前記表示手段としてのネットワークに接続されたクライアント端末に対し前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする。

#### 【0016】

この請求項5にかかる発明によれば、プログラム入力手段によって、表示手段としてのネットワークに接続されたクライアント端末に対し入力画面を表示し、入力画面から前記カスタマイズプログラムを入力させることで、カスタマイズプログラムの入力を当該入力が容易なクライアント端末で行うことができ、画像形成装置上でカスタマイズプログラムの作成編集を行う場合に比べて、操作表示部のカスタマイズをより容易に行うことができ、操作表示部のカスタマイズを効率的に行うことができる。

#### 【0017】

また、請求項6にかかる発明は、請求項1～4のいずれか一つに記載の画像形成装置において、前記プログラム入力手段は、前記表示手段としての前記オペレ

ーションパネルの操作表示部に前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする。

#### 【0018】

この請求項6にかかる発明によれば、プログラム入力手段によって、表示手段としてのオペレーションパネルの操作表示部に入力画面を表示し、入力画面から前記カスタマイズプログラムを入力させることで、ネットワークに接続されていないスタンドアローンの画像形成装置においても操作表示部のカスタマイズを行うことができる。

#### 【0019】

また、請求項7にかかる発明は、請求項1～6のいずれか一つに記載の画像形成装置において、画像形成装置の起動時にアプリケーションプログラムを起動する起動手段をさらに備え、前記起動手段は、前記プログラム実行手段を起動することを特徴とする。

#### 【0020】

この請求項7にかかる発明によれば、画像形成装置の起動時にアプリケーションプログラムを起動する起動手段によってプログラム実行手段を起動することで、プログラム実行手段の起動を画像形成装置の起動処理の一環として自動的に行うことができ、操作表示部のカスタマイズの検証をより効率的に行うことができる。

#### 【0021】

また、請求項8にかかる発明は、請求項1～7のいずれか一つに記載の画像形成装置において、前記プログラム実行手段は、汎用OSの一部として動作するインタプリタであることを特徴とする。

#### 【0022】

この請求項8にかかる発明によれば、プログラム実行手段汎用OSの一部として動作するインタプリタであるため、別途インタプリタを画像形成装置にインストールすることなく、カスタマイズプログラムの実行を行うことができ、操作表示部のカスタマイズを容易に行うことができる。

#### 【0023】

また、請求項 9 にかかる発明は、請求項 1 ~ 7 のいずれか一つに記載の画像形成装置において、前記プログラム実行手段は、汎用 OS の配下で動作するインタプリタであることを特徴とする。

#### 【0024】

この請求項 9 にかかる発明によれば、プログラム実行手段は汎用 OS の配下で動作するインタプリタであるため、汎用 OS で提供されるインタプリタよりも高度な処理を行えるインタプリタを利用することができ、操作表示部のカスタマイズの範囲を拡張することができる。

#### 【0025】

請求項 10 にかかる発明は、請求項 1 ~ 7 のいずれか一つに記載の画像形成装置において、前記画像形成装置は、画像形成処理で使用されるハードウェア資源と、画像形成処理にかかるユーザサービスにそれぞれ固有の処理を行うアプリケーションと、前記アプリケーションと前記ハードウェア資源との間に介在するコントロールサービスとを更に有し、前記プログラム実行手段及び前記カスタマイズプログラムは、前記アプリケーションとして動作することを特徴とする。

#### 【0026】

この請求項 10 にかかる発明によれば、プログラム実行手段をアプリケーションとして動作するインタプリタとすることができます。

#### 【0027】

また、請求項 11 に記載の発明によれば、請求項 10 の記載において、前記カスタマイズプログラムは、前記プログラム実行手段により、前記コントロールサービスに対する関数呼び出しを行うこととするので、種々のアプリケーションを既存の関数を用いて作成することが可能となる。

#### 【0028】

また、請求項 12 にかかる発明は、画面表示および操作入力を行う操作表示部を有するオペレーションパネルを備えた画像形成装置であって、利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させるプログ

ラム入力手段と、前記カスタマイズプログラムを、前記操作表示部に表示されるとともに入力を行うキーに割り付ける割付手段と、前記操作表示部から前記割付手段によって割り付けられたキーの操作入力があった場合に、前記プログラム入力手段によって入力されたカスタマイズプログラムをステップごとに逐次実行するプログラム実行手段とを備えたことを特徴とする。

#### 【0029】

この請求項12にかかる発明によれば、プログラム入力手段によって、表示手段に対し前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示してカスタマイズプログラムを入力させ、割付手段によって、カスタマイズプログラムを操作表示部に表示されるとともに入力を行うキーに割り付けて、プログラム実行手段によって、操作表示部から割り付けられたキーの操作入力があった場合に入力されたカスタマイズプログラムをステップごとに逐次実行することで、操作表示部に表示されるキーの操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができる。

#### 【0030】

また、請求項13にかかる発明は、請求項12に記載の画像形成装置において、前記割付手段は、画像形成装置の起動時に前記カスタマイズプログラムを、前記キーに割り付けることを特徴とする。

#### 【0031】

この請求項13にかかる発明によれば、割付手段によって、画像形成装置の起動時にカスタマイズプログラムをキーに割り付けることで、カスタマイズプログラムが実行される場合のキー操作を画像形成装置の起動時に動的に定めることができ、操作表示部のカスタマイズを柔軟的に行うことができる。

#### 【0032】

また、請求項14にかかる発明は、請求項13に記載の画像形成装置において、前記割付手段は、前記キーと前記カスタマイズプログラムの割り付けを設定するキー割付け画面を表示し、前記キー割付け画面からの設定内容に基づいて前記カスタマイズプログラムを前記キーに割り付けることを特徴とする。

**【0033】**

この請求項14にかかる発明によれば、割付手段によって、キーとカスタマイズプログラムの割り付けを設定するキー割付け画面を表示し、キー割付け画面からの設定内容に基づいてカスタマイズプログラムをキーに割り付けることで、キーとカスタマイズプログラムの動的な割り付けを簡易なユーザインターフェースで実現することができる。

**【0034】**

また、請求項15にかかる発明は、画像形成装置における画面表示および操作入力を行うオペレーションパネルの操作表示部をカスタマイズする操作表示部プログラム生成方法であって、利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示し、前記入力成画面から前記カスタマイズプログラムを入力させるプログラム入力ステップと、前記プログラム入力ステップによって入力されたカスタマイズプログラムをステップごとに逐次実行するプログラム実行ステップと、を含むことを特徴とする。

**【0035】**

この請求項15にかかる発明によれば、プログラム入力ステップによって、利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示して、この入力画面からカスタマイズプログラムを入力させ、プログラム実行ステップによって、入力されたカスタマイズプログラムをステップごとに逐次実行することで、操作表示部のカスタマイズプログラムの開発において、表示手段の入力画面から追加、修正ができ、また追加修正による再コンパイル、再リンクなしに画像形成装置で実行でき、操作表示部のカスタマイズを容易に行うことができる。

**【0036】**

また、請求項16にかかる発明は、請求項15に記載の操作表示部プログラム生成方法において、前記プログラム入力ステップは、さらに前記入力画面から入力されたカスタマイズプログラムを、記憶手段の予め定められたカスタマイズ領

域に格納し、前記プログラム実行ステップは、前記カスタマイズ領域に格納された前記カスタマイズプログラムをステップごとに逐次実行することを特徴とする。

#### 【0037】

この請求項16にかかる発明によれば、プログラム入力ステップによって、入力画面から入力されたカスタマイズプログラムを、記憶手段の予め定められたカスタマイズ領域に格納し、前記プログラム実行ステップによって、カスタマイズ領域に格納されたカスタマイズプログラムをステップごとに逐次実行することで、予めカスタマイズ領域を定め当該カスタマイズ領域にカスタマイズプログラムを格納しておけば、プログラム実行時に実行するカスタマイズプログラムを容易に特定することができ、操作表示部のカスタマイズおよびその検証を容易に行うことができる。

#### 【0038】

また、請求項17にかかる発明は、請求項15または16に記載の操作表示部プログラム生成方法において、前記プログラム入力ステップは、さらに前記操作表示部に表示されるとともに操作入力を行うキーとカスタマイズプログラムとの対応付けを行うことを特徴とする。

#### 【0039】

この請求項17にかかる発明によれば、プログラム入力ステップは、さらに前記操作表示部に表示されるとともに操作入力を行うキーとカスタマイズプログラムとの対応付けを行うことで、キー操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができる。

#### 【0040】

また、請求項18にかかる発明は、請求項15～17のいずれか一つに記載の操作表示部プログラム生成方法において、前記プログラム入力ステップは、さらに入力されたカスタマイズプログラムを動作確認のために実行させることを特徴とする。

#### 【0041】

この請求項 18 にかかる発明によれば、プログラム入力ステップによって、入力されたカスタマイズプログラムを動作確認のために実行させることで、カスタマイズプログラムを実際に画像形成装置で動作させる前に、表示手段上で簡単な検証を行うことができ、操作表示部のカスタマイズを効率的に行うことができる。

#### 【0042】

また、請求項 19 にかかる発明は、請求項 15～18 のいずれか一つに記載の操作表示部プログラム生成方法において、前記プログラム入力ステップは、前記表示手段としてのネットワークに接続されたクライアント端末に対し前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする。

#### 【0043】

この請求項 19 にかかる発明によれば、プログラム入力ステップによって、表示手段としてのネットワークに接続されたクライアント端末に対し入力画面を表示し、入力画面からカスタマイズプログラムを入力させることで、カスタマイズプログラムの入力を当該入力が容易なクライアント端末で行うことができ、画像形成装置上でカスタマイズプログラムの作成編集を行う場合に比べて、操作表示部のカスタマイズをより容易に行うことができ、操作表示部のカスタマイズを効率的に行うことができる。

#### 【0044】

また、請求項 20 にかかる発明は、請求項 15～18 のいずれか一つに記載の操作表示部プログラム生成方法において、前記プログラム入力ステップは、前記表示手段としての前記オペレーションパネルの操作表示部に前記入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させることを特徴とする。

#### 【0045】

この請求項 20 にかかる発明によれば、プログラム入力ステップによって、表示手段としてのオペレーションパネルの操作表示部に入力画面を表示し、入力画面からカスタマイズプログラムを入力させることで、ネットワークに接続されて

いないスタンドアローンの画像形成装置においても操作表示部のカスタマイズを行うことができる。

#### 【0046】

また、請求項21にかかる発明は、請求項15～20のいずれか一つに記載の操作表示部プログラム生成方法において、画像形成装置の起動時にアプリケーションプログラムを起動する起動ステップをさらに含み、前記起動ステップは、前記プログラム実行ステップを実行するプログラムを起動することを特徴とする。

#### 【0047】

この請求項21にかかる発明によれば、画像形成装置の起動時にアプリケーションプログラムを起動する起動ステップによって、プログラム実行ステップを実行するプログラムを起動することで、プログラム実行ステップを実行するプログラムの起動を画像形成装置の起動処理の一環として自動的に行うことができ、操作表示部のカスタマイズの検証をより効率的に行うことができる。

#### 【0048】

また、請求項22にかかる発明は、画像形成装置における画面表示および操作入力を行うオペレーションパネルの操作表示部をカスタマイズする操作表示部プログラム生成方法であって、利用者に対する画面表示および入力を行わせる表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示し、前記入力画面から前記カスタマイズプログラムを入力させるプログラム入力ステップと、前記カスタマイズプログラムを、前記操作表示部に表示されるとともに入力を行うキーに割り付ける割付ステップと、前記操作表示部から前記割付ステップによって割り付けられたキーの操作入力があった場合に、前記プログラム入力ステップによって入力されたカスタマイズプログラムをステップごとに逐次実行するプログラム実行ステップと、を含むことを特徴とする。

#### 【0049】

この請求項22にかかる発明によれば、プログラム入力ステップによって、表示手段に対し、前記操作表示部からの入力に対する実行処理をインタプリタ用言語で記述したカスタマイズプログラムの入力画面を表示して、入力画面から前記

カスタマイズプログラムを入力させ、割付ステップによって、カスタマイズプログラムを操作表示部に表示されるとともに入力を行うキーに割り付け、プログラム入力ステップによって、操作表示部から割り付けられたキーの操作入力があった場合に、入力されたカスタマイズプログラムをステップごとに逐次実行することで、操作表示部に表示されるキーの操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができる。

#### 【0050】

また、請求項23にかかる発明は、請求項22に記載の操作表示部プログラム生成方法において、前記割付ステップは、画像形成装置の起動時に前記カスタマイズプログラムを、前記キーに割り付けることを特徴とする。

#### 【0051】

この請求項23にかかる発明によれば、割付ステップによって、画像形成装置の起動時にカスタマイズプログラムをキーに割り付けることで、カスタマイズプログラムが実行される場合のキー操作を画像形成装置の起動時に動的に定めることができ、操作表示部のカスタマイズを柔軟的に行うことができる。

#### 【0052】

また、請求項24にかかる発明は、請求項23に記載の操作表示部プログラム生成方法において、前記割付ステップは、前記キーと前記カスタマイズプログラムの割り付けを設定するキー割付け画面を表示し、前記キー割付け画面からの設定内容に基づいて前記カスタマイズプログラムを前記キーに割り付けることを特徴とする。

#### 【0053】

この請求項24にかかる発明によれば、割付ステップによって、キーとカスタマイズプログラムの割り付けを設定するキー割付け画面を表示し、キー割付け画面からの設定内容に基づいてカスタマイズプログラムをキーに割り付けることで、キーとカスタマイズプログラムの動的な割り付けを簡易なユーザインターフェースで実現することができる。

#### 【0054】

### 【発明の実施の形態】

以下に添付図面を参照して、この発明にかかる画像形成装置および操作表示部プログラム生成方法の好適な実施の形態を詳細に説明する。

#### 【0055】

##### (実施の形態 1)

図1は、この発明の実施の形態1である画像形成装置（以下、「複合機」という）の主要部の構成とネットワーク構成を示すブロック図である。実施の形態1にかかる複合機100は、インターネットに接続され、当該インターネットに接続されたPC（Personal Computer）などのクライアント端末から複合機の操作を行うオペレーションパネルの操作表示部のカスタマイズプログラムの入力およびデバッグ作業を行うことを可能としたものである。

#### 【0056】

図1に示すように、複合機100とPC200は、インターネット220で接続されており、通信プロトコルとしてTCP/IPを利用している。複合機100は、複合機初期化部129と、プログラム起動部131と、プログラミングサービス132と、インタプリタ134と、NCS（ネットワークコントロールサービス）128と、httpd（http デーモン）106と、共有メモリ105と、HDD103とを備えた構成となっている。ここで、プログラミングサービス132は、本発明におけるプログラム入力手段を構成し、インタプリタ134は、本発明におけるプログラム実行手段を構成する。また、プログラム起動手段131は、本発明における起動手段を構成する。

#### 【0057】

プログラミングサービス132は、ネットワークに接続されたクライアント端末としてのPC200のWEBブラウザに後述するhttpファイルで構成されたWEBページであるプログラミング画面201を表示し、このプログラミング画面201からオペレーションパネル210の操作表示部のカスタマイズプログラムを入力させるものである。また、プログラミングサービス132は、PCから入力されたカスタマイズプログラムをHDD103のカスタマイズ領域として作成されたカスタマイズディレクトリ212に保存する。さらに、プログラミン

グサービス132は、複合機100上でカスタマイズプログラムが実行されるよう、後述する起動設定ファイル211に入力されたカスタマイズプログラムを登録する。

#### 【0058】

ここで、プログラミングサービス132とPC200のWEBブラウザの間のプログラミング画面などのhttpファイルの転送は、httpプロトコルに従って行われる。このため、プログラミングサービス132は、Webサーバ（httpサーバ）としての役割も担っている。

#### 【0059】

カスタマイズプログラムは、インタプリタで解釈可能な言語で記述され、具体的には、汎用OSはUNIX（登録商標）の場合、シェルスクリプトなどである。

#### 【0060】

インタプリタ134は、カスタマイズプログラムをステップごとに解析し逐次実行するものであり、本実施の形態では、シェルスクリプトを実行するシェル（bsh, cshなど）である。このシェル134は、汎用OS121としてのUNIX（登録商標）にOSのカーネル135とは別個に存在するものである。

#### 【0061】

プログラム起動部131は、本発明における外部アプリ起動手段を構成し、HDD103の診断結果が正常である場合に、HDD103にインストールされている外部アプリ117を起動するものである。また、プログラム起動部131は、ICカードなどの記憶媒体に保存された外部アプリ117を起動する。

#### 【0062】

複合機初期化部129は、汎用OS121の上で最初に起動されるプロセスであり、コントロールサービスやアプリケーション130（外部アプリ117を除く）の起動およびプログラム起動部131の起動を行うものである。

#### 【0063】

NCS128は、ネットワークを制御するものであり、httpd106から通知されたリクエストメッセージ受信の旨をプログラミングサービス132に通

知する。

#### 【0064】

h t t p d 1 0 6 は、OSに含まれるプロセス（デーモン）であり、ポート80番を常時監視してリクエストメッセージの受信を行うとともに、レスポンスマッセージの送信を行うものである。なお、リクエストメッセージおよびレスポンスマッセージの構造は、通常のh t t p プロトコルにおける各メッセージの構造と同様であり、各メッセージには、h t m l 形式で記述されたメッセージボディが含まれている。また、h t t p d 1 0 6 は、インターネット220経由で受信したリクエストメッセージ受信の旨をN C S 1 2 8 に通知するとともに、リクエストメッセージを共有メモリ105に格納する。

#### 【0065】

この共有メモリ105は、h t t p d 1 0 6 とプログラミングサービス132とのプロセス間通信に利用されるものであり、共有メモリ105を介してリクエストメッセージ、レスポンスマッセージなどの受け渡しを行う。

#### 【0066】

H D D 1 0 3 には、起動設定ファイル211とが格納され、またH D D 1 0 3 のカスタマイズディレクトリ212にカスタマイズプログラムが格納されるようになっている。

#### 【0067】

P C 2 0 0 は、複合機100のプログラミングサービス132をサーバとしたクライアント端末であり、P C 2 0 0 で実行されるW E B ブラウザにプログラミングサービス132からh t t p プロトコルで送信されたプログラミング画面201やその他画面を表示し、操作入力を行えるようになっている。

#### 【0068】

次に、本実施の形態にかかる複合機100の全体の機能的構成について説明する。図2は、実施の形態1の複合機100の構成を示すブロック図である。図1に示すように、複合機100は、白黒ラインプリンタ（B&W LP）101と、カラーラインプリンタ（Color LP）102と、ハードディスク装置（H D D）103と、スキャナ、ファクシミリ、メモリ、ネットワークインターフェースなどのハ

ドウェアリソース104を有するとともに、プラットホーム120と、アプリケーション130と、複合機初期化部129と、プログラム起動部131と、プログラミングサービス132とから構成されるソフトウェア群110とを備えている。

#### 【0069】

プラットホーム120は、アプリケーションからの処理要求を解釈してハードウェア資源の獲得要求を発生させるコントロールサービスと、一または複数のハードウェア資源の管理を行い、コントロールサービスからの獲得要求を調停するシステムリソーススマネージャ（SRM）123と、汎用OS121とを有する。

#### 【0070】

コントロールサービスは、複数のサービスモジュールから形成され、SCS（システムコントロールサービス）122と、ECS（エンジンコントロールサービス）124と、MCS（メモリコントロールサービス）125と、OCS（オペレーションパネルコントロールサービス）126と、FCS（ファックスコントロールサービス）127と、NCS（ネットワークコントロールサービス）128とから構成される。また、プログラミングサービス132も、コントロールサービス層に含まれている。なお、このプラットホーム120は、あらかじめ定義された関数により前記アプリケーション130から処理要求を受信可能とするアプリケーションプログラムインターフェース（API）を有する。

#### 【0071】

汎用OS121は、UNIX（登録商標）などの汎用オペレーティングシステムであり、プラットホーム120並びにアプリケーション130の各ソフトウェアをそれぞれプロセスとして並列実行する。

#### 【0072】

SRM123のプロセスは、SCS122とともにシステムの制御およびリソースの管理を行うものである。SRM123のプロセスは、スキャナ部やプリンタ部などのエンジン、メモリ、HDDファイル、ホストI/O（セントロI/F、ネットワークI/F、IEEE1394 I/F、RS232C I/Fなど）のハードウェア資源を利用する上位層からの要求にしたがって調停を行い、実

行制御する。

#### 【0073】

具体的には、この S R M 1 2 3 は、要求されたハードウェア資源が利用可能であるか（他の要求により利用されていないかどうか）を判断し、利用可能であれば要求されたハードウェア資源が利用可能である旨を上位層に伝える。また、 S R M 1 2 3 は、上位層からの要求に対してハードウェア資源の利用スケジューリングを行い、要求内容（例えば、プリンタエンジンにより紙搬送と作像動作、メモリ確保、ファイル生成など）を直接実施している。

#### 【0074】

S C S 1 2 2 のプロセスは、アプリ管理、操作部制御、システム画面表示、 L E D 表示、リソース管理、割り込みアプリ制御などを行う。

#### 【0075】

E C S 1 2 4 のプロセスは、白黒ラインプリンタ（B&W LP）101、カラーラインプリンタ（Color LP）102、スキャナ、ファクシミリなどからなるハードウェアリソース103のエンジンの制御を行う。

#### 【0076】

M C S 1 2 5 のプロセスは、画像メモリの取得および解放、ハードディスク装置（HDD）の利用、画像データの圧縮および伸張などを行う。

#### 【0077】

F C S 1 2 7 のプロセスは、システムコントローラの各アプリ層から P S T N ／ I S D N 網を利用したファクシミリ送受信、 B K M （バックアップ S R A M ）で管理されている各種ファクシミリデータの登録／引用、ファクシミリ読みとり、ファクシミリ受信印刷、融合送受信を行うための A P I を提供する。

#### 【0078】

N C S 1 2 8 のプロセスは、ネットワーク I ／ O を必要とするアプリケーションに対して共通に利用できるサービスを提供するためのプロセスであり、ネットワーク側から各プロトコルによって受信したデータを各アプリケーションに振り分けたり、アプリケーションからデータをネットワーク側に送信する際の仲介を行う。具体的には、 I W S G 、 O S G 、 V Q P S G 、 W H O Q H W G 、 V P W S G などのサーバデータモ

ンや、同プロトコルのクライアント機能などを有している。

#### 【0079】

OCS126のプロセスは、オペレータ（ユーザ）と本体制御間の情報伝達手段となるオペレーションパネル（操作パネル）210の制御を行う。OCS126は、オペレーションパネルからキー押下（またはタッチ操作）をキーイベントとして取得し、取得したキーに対応したキーイベント関数をSCS122に送信するOCSプロセスの部分と、アプリケーション130またはコントロールサービスからの要求によりオペレーションパネルに各種画面を描画出力する描画関数やその他オペレーションパネルに対する制御を行う関数などがあらかじめ登録されたOCSライブラリの部分とから構成される。このOCSライブラリは、アプリケーション130およびコントロールサービスの各モジュールにリンクされて実装されている。なお、OCS126のすべてをプロセスとして動作させるように構成しても良く、あるいはOCS126のすべてをOCSライブラリとして構成しても良い。

#### 【0080】

アプリケーション130は、ページ記述言語（PDL）、PCLおよびポストスクリプト（PS）を有するプリンタ用のアプリケーションであるプリンタアプリ111と、コピー用アプリケーションであるコピーアプリ112と、ファクシミリ用アプリケーションであるファックスアプリ113と、スキャナ用アプリケーションであるスキャナアプリ114と、ネットワークファイル用アプリケーションであるネットファイルアプリ115と、工程検査用アプリケーションである工程検査アプリ116とを有している。

#### 【0081】

アプリケーション130の各プロセス、コントロールサービスの各プロセスは、関数呼び出しとその戻り値送信およびメッセージの送受信によってプロセス間通信を行いながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを実現している。

#### 【0082】

このように、実施の形態1にかかる複合機100には、複数のアプリケーショ

ン130および複数のコントロールサービスが存在し、いずれもプロセスとして動作している。そして、これらの各プロセス内部には、一または複数のスレッドが生成されて、スレッド単位の並列実行が行われる。そして、コントロールサービスがアプリケーション130に対し共通サービスを提供しており、このため、これらの多数のプロセスが並列動作、およびスレッドの並列動作を行って互いにプロセス間通信を行って協調動作をしながら、コピー、プリンタ、スキャナ、ファクシミリなどの画像形成処理にかかるユーザサービスを提供するようになっている。

#### 【0083】

また、複合機100には、複合機100の顧客、サードベンダなどの第三者がコントロールサービス層の上のアプリケーション層に外部アプリを開発して搭載することが可能となっている。

#### 【0084】

なお、実施の形態1にかかる複合機100では、複数のアプリケーション130のプロセスと複数のコントロールサービスのプロセスとが動作しているが、アプリケーション130とコントロールサービスのプロセスがそれぞれ単一の構成とすることも可能である。また、各アプリケーション130は、アプリケーションごとに追加または削除することができる。

#### 【0085】

インストーラ118は、第三者が開発した外部アプリ117を、HDD103にインストールするものである。本実施の形態にかかる複合機100では、フラッシュカードなどのICカードの記憶媒体に外部アプリ117を格納し、インストーラ118によって外部アプリ117をHDD103にインストールし、プログラム起動部131によって、HDD103から外部アプリ117を起動してアプリケーション層で動作させるようになっている。一方、プリンタアプリ111、コピーアプリ112、ファックスアプリ113、スキャナアプリ114、ネットファイルアプリ115、工程検査アプリ116などの複合機100の出荷時に提供されるアプリケーション130、各コントロールサービス、HDD診断部132およびプログラム起動部131は、フラッシュメモリに出荷時に組み込まれ

ており、複合機100の起動時（電源投入時）に複合機初期化部129によって起動されるようになっている。

#### 【0086】

次に、以上のように構成された本実施の形態にかかる複合機100によるオペレーションパネル210の操作表示部のカスタマイズ方法について説明する。まず、利用者は、PC200でWEBブラウザを起動し、複合機100上にあるプログラミング画面のWEBページのURL（例えば、`http://www.xxx.yyy/zzz/debug.htm`）を指定して複合機100のプログラミングサービス132にアクセスする。WEBブラウザからのアクセスを受けたプログラミングサービス132は、プログラミング画面201（`debug.htm`）をPC200のWEBブラウザに表示させる。

#### 【0087】

図6は、プログラミング画面201の内容の一例を示す説明図である。図6に示すように、プログラミング画面201には、プログラム入力フィールドと、参照ボタンと、アップロードボタンと、保存ボタンと、消去ボタンと、テスト実行ボタンと、実行設定ボタンが表示される。

#### 【0088】

ここで、プログラミング画面201は、`htm1`形式（Hyper Text Markup Language）で記述されたファイル（`debug.htm`）である。当該ファイルをPC200上でWEBブラウザによって表示することにより、プログラミング画面201がPC200で表示される。また、プログラミングサービス132は、プログラミング画面（`debug.htm`）からの操作入力や要求に応じたカスタマイズプログラムの入力に関する処理等を、CGI（Common Gateway Interface）のスクリプトを実行し、その処理結果をプログラミング画面（`debug.htm`）に反映して、PC200のWEBブラウザに返すようになっている。具体的には、プログラミングサービス132は、カスタマイズプログラムの入力処理等の要求（ボタンのクリック操作等）があった場合には、`debug.cgi`のスクリプトを実行する。なお、本実施の形態では、CGIスクリプトを実行することにより、WEBアプリ1

117による各処理を実行しているが、CGIスクリプトから呼び出される別のプログラムで実行するように構成しても良い。なお、プログラミング画面を、X<sub>ML</sub>形式（eXtensible Markup Language）で記述するように構成することも可能である。

#### 【0089】

参照ボタンは、ブラウザが動作しているPC200の記憶領域に格納されたカスタマイズプログラムを参照するためのボタンであり、この参照ボタンを押下すると、PC200の記憶領域内のディレクトリが表示され、所望のディレクトリに存在するカスタマイズプログラムを指定すると、ファイル名フィールドに指定されたカスタマイズプログラムのファイル名が表示される。なお、参照ボタンを押下したときに、複合機100のHDD103などの記憶媒体のディレクトリを参照するように構成しても良い。

#### 【0090】

アップロードボタンは、参照ボタンで指定したカスタマイズプログラムあるいは入力フィールドに入力したカスタマイズプログラムを、複合機100に転送（アップロード）するためにクリック操作するものである。

#### 【0091】

プログラム入力フィールドは、カスタマイズプログラムを入力する領域である。このプログラム入力フィールドには、PC200のキーボードなどの入力装置からカスタマイズプログラムを直接入力する他、上述のように参照ボタンをクリック操作することによってPC200の所望のディレクトリに存在する既存のプログラムファイルを指定することによってカスタマイズプログラムを入力する。`debug.htm`において、このプログラム入力フィールドに相当するすべての行の先頭には、「`program_list:`」の文字列が記述されている。

#### 【0092】

保存ボタンは、入力したカスタマイズプログラムを複合機100にアップロードした後に、カスタマイズディレクトリに格納する場合にクリック操作するものである。消去ボタンは、入力フィールドに入力されたカスタマイズプログラムを

消去する際にクリック操作するものである。

#### 【0093】

実行設定ボタンは、カスタマイズディレクトリに保存されたカスタマイズプログラムを複合機100で実行できるように、カスタマイズプログラムとカスタマイズプログラムを実行させるボタンを対応づける際にクリック操作するものである。この実行設定ボタンが押下されると、プログラミングサービス132は、カスタマイズプログラム中で指定されたボタン（キー）を、カスタマイズディレクトリに存在する画面ファイル中のボタン（キー）からキーコードに基づいて検索し、カスタマイズプログラムとボタン（キー）との対応付けを行い、カスタマイズプログラムの名称とボタン（キー）のキーコードのキー対応付けテーブルを生成してHDD103またはフラッシュメモリなどに当該テーブルを格納する。

#### 【0094】

テスト実行ボタンは、入力されたカスタマイズプログラムを試験的に実行する際にクリック操作するものである。

#### 【0095】

図7は、入力するカスタマイズプログラムの内容の一例を示す説明図である。図7に示すように、本実施の形態では、UNIX（登録商標）のシェルスクリプトを入力するようになっている。入力されたシェルスクリプトは、汎用OS121（UNIX（登録商標））のコマンド・インタプリタであるシェル134によってステップごとに逐次実行され、このため、入力したカスタマイズプログラムのコンパイルおよびリンクは不要となっている。

#### 【0096】

また、シェルスクリプトには、開発ライブラリで提供される複合機専用コマンドを指定できるようになっている。図7のスクリプトは、原稿をスキャナで読み込み（scaniimage）、読み込み時にTIFF形式に圧縮された画像データを伸長し（decomp）、伸長した画像データに対し文字認識処理を行う（ocr）。そして、文字認識処理による認識結果をテキストデータとしてメール送信（mail）する例を示している。このスクリプトの中では、scaniimage, decomp, ocrの各コマンドが開発ライブラリで提供されるコマン

ドである。また、カスタマイズプログラムには、このシェルスクリプトを実行するためのボタン（キー）の指定が可能となっている。

#### 【0097】

PC200のWEBブラウザに表示されているプログラミング画面201（debug.htm）において、参照ボタンをクリックしてPC200に格納されているカスタマイズプログラムを指定し、あるいはプログラム入力フィールドに直接カスタマイズプログラムを入力し、アップロードボタンをクリック操作した場合、WEBブラウザから複合機100のプログラミングサービス132に対しリクエストメッセージが送信される。このリクエストメッセージのメッセージボディにはdebug.htmが含められており、このdebug.htmには、アップロードボタンをクリック操作したときにdebug.cgiを呼び出す旨が記述されている。このためリクエストメッセージを受信して解析を行ったプログラミングサービス132は、debug.cgiを実行する。

#### 【0098】

ここで、リクエストメッセージを受信した複合機100における処理について説明する。図3は、リクエストメッセージを受信した複合機100における処理の手順を示すフローチャートである。

#### 【0099】

httpd106は、ポート80番を常時監視しており、PC200からリクエストメッセージをポート80番で受信する（ステップS301）。そして、httpd106は、受信したリクエストメッセージを読み出し、リクエストメッセージを共有メモリ105に書き込む（ステップS302）。次に、httpd106は、リクエストメッセージ受信の旨をNCS128に通知する（ステップS303）。

#### 【0100】

通知を受けたNCS128は、さらにリクエストメッセージ受信の旨をプログラミングサービス132に通知する（ステップS304）。プログラミングサービス132は、リクエストメッセージ受信の旨の通知を受けたことをトリガとして、共有メモリ105を参照して、リクエストメッセージを読み出す（ステップ

S305)。そして、リクエストメッセージの内容を解析して（ステップS306）、リクエストメッセージのメッセージボディに記述された内容に応じた処理を実行する（ステップS307）。このとき、メッセージボディのhtml記述にCGIを実行する旨が記述されていれば、対応するCGIを実行する。

#### 【0101】

そして、プログラミングサービス132は、処理の実行結果をレスポンスマッセージとして、httpd106経由でリクエストメッセージの送信元であるPC200に返信する（ステップS308）。

#### 【0102】

次に、プログラミングサービス132によるdebug.cgiの処理について説明する。図4は、プログラミングサービス132によるdebug.cgiの処理の手順を示すフローチャートである。プログラミングサービス132は、指定されたカスタマイズプログラムのプログラムファイルを複合機100にアップロードする。あるいは受信したリクエストメッセージの中のdebug.htmのプログラム入力フィールドに直接入力されたカスタマイズプログラムをプログラムファイルにした後、このプログラムファイルを複合機100にアップロードする（ステップS401）。

#### 【0103】

次に、プログラミングサービス132は、debug.htmを先頭から一行づつ読み込む（ステップS402）。そして、読み込んだ行が「program\_list:」の文字列を含むか否かを調べることにより、プログラム入力フィールドか否かを判断する（ステップS403）。

#### 【0104】

そして、プログラム入力フィールドであると判断した場合には、アップロードされたプログラムファイルを読み込んで（ステップS404）、プログラムファイルの内容、すなわちカスタマイズプログラムをPC200に送信する（ステップS405）。

#### 【0105】

一方、ステップS403において、プログラム入力フィールドでないと判断し

た場合には、ステップS402で読み込んだ一行をそのままPC200に送信する（ステップS406）。

#### 【0106】

そして、ステップS402からS406までの処理を、debug.htmのすべての行について繰り返し行う（ステップS407）。これにより、PC200のWEBブラウザには、debug.cgiによる処理結果のdebug.htmが順次表示されることになる。

#### 【0107】

このようにプログラミング画面201から入力されたカスタマイズプログラムが複合機100にアップロードされ、プログラム入力フィールドにアップロード後のカスタマイズプログラムが表示された後、保存ボタン、実行設定ボタン、テスト実行ボタンのクリック操作が可能となる。これらのいずれかのボタンをクリック操作した場合におけるプログラミングサービス132の処理について説明する。

#### 【0108】

図5は、プログラミング画面201において、保存ボタン、実行設定ボタン、テスト実行ボタンのいずれかのクリック操作が行われた場合のプログラミングサービス132の処理の手順を示すフローチャートである。かかる処理は、プログラミング画面201のhtmlファイルにJava（登録商標）Scriptで記述されている。

#### 【0109】

まず、プログラミングサービス132は、クリック操作されたボタンの種類を判断する（ステップS501）。プログラミング画面201で保存ボタンがクリックされた場合には、プログラミングサービス132は、ファイル名称を利用者に指定させ、カスタマイズプログラムを指定された名称で複合機100のHDD103のカスタマイズディレクトリに格納する（ステップS502）。

#### 【0110】

図8は、カスタマイズディレクトリに格納されたカスタマイズプログラムの名称の一例を示す説明図である。図8に示すように、複合機100のHDD103

のカスタマイズディレクトリは、`hdd/xxx/opepane`となっており、このカスタマイズディレクトリにカスタマイズプログラムである`shell1`, `shell2`, `shell3`が格納されている。

#### 【0111】

一方、プログラミング画面201で実行設定ボタンが押下された場合には、プログラミングサービス132は、上述のようにカスタマイズプログラムとボタンとの対応付けを行う（ステップS503）。

#### 【0112】

図9は、キー対応付けテーブルの内容の一例を示す説明図である。図9に示すように、キー対応付けテーブルには、ボタンのキーコードとカスタマイズプログラム名称が対応付けられている。

#### 【0113】

また、プログラミング画面201でテスト実行ボタンが押下された場合には、プログラミングサービス132は、入力されたカスタマイズプログラムの実行を行う（ステップS504）。このとき、テスト実行によって、オペレーションパネル210に表示される画面と同一の画面をPC200で動作しているブラウザに表示する。

#### 【0114】

次に、このようにして生成されたカスタマイズプログラムの実行処理について説明する。図10は、複合機100の電源投入が行われてから、カスタマイズプログラムの実行が可能となるまでの一連の処理の手順を示すフローチャートである。

#### 【0115】

カスタマイズプログラムのカスタマイズディレクトリへの保存および起動設定ファイル211への設定が完了したら、利用者は複合機100の電源を再投入する。このとき複合機100では、ROMモニタ（図示せず）によってハードウェアの診断処理を行い（ステップS1001）、その後汎用OS121を起動し（ステップS1002）、さらに複合機初期化部129を起動する（ステップS1003）。複合機初期化部129は、まずコントロールサービスの起動を行い

(ステップS1004)、続いてコピーアプリ112、プリンタアプリ111などの複合機100の出荷時に提供されている既存アプリを起動し(ステップS1005)、さらに出荷後新たに開発された外部アプリなどのプログラムを起動するためにプログラム起動部131を起動する(ステップS1006)。

#### 【0116】

プログラム起動部131は、まずHDD103の起動設定ファイル211を参考し(ステップS1007)、起動すべきプログラムの名称を取得する。

#### 【0117】

図11は、起動設定ファイル211の一例を示す説明図である。図11に示すように、起動設定ファイル211には、「プログラム名称カスタマイズプログラム名」の形式で、プログラム起動部131によって起動されるプログラムが登録されている。図11の例では、シェルスクリプトであるシェルのプログラムが最初に登録され、その後に、外部アプリとしてxxxアプリが登録されている。

#### 【0118】

このため、図11の例では、プログラム起動部131は、まずシェルを起動し、次いで、外部アプリであるXXXアプリを順に起動するように設定されている。このため、プログラム起動部131は、まずインタプリタ134(シェル)を起動し(ステップS1008)、次いで、外部アプリであるXXXアプリを順に起動する(ステップS1009)。

#### 【0119】

従って、オペレーションパネル210の操作表示部に表示されたボタン(キー)の中で、カスタマイズプログラムが割り付けられたボタン(キー)を押下(タッチ操作)すると、シェルスクリプトで記述されたカスタマイズプログラムがシェルによって実行されることになる。

#### 【0120】

このように実施の形態1にかかる複合機100では、プログラミングサービス132によって、ネットワーク上のPC200に、オペレーションパネル210の操作表示部のキー操作入力に対する実行処理をインタプリタ用言語であるシェルスクリプトで記述したカスタマイズプログラムのプログラミング画面を表示し

、この画面からカスタマイズプログラムを入力させ、複合機100のシェル134によって、入力されたカスタマイズプログラムをステップごとに逐次実行することで、操作表示部のカスタマイズプログラムの開発において、プログラミング画面から追加、修正ができる、また追加修正による再コンパイル、再リンクなしに複合機100で実行でき、操作表示部のカスタマイズを容易に行うことができる。

#### 【0121】

また、実施の形態1にかかる複合機100では、プログラミングサービス132によって、インターネットに接続されたPC200に対しプログラミング画面を表示し、この画面からカスタマイズプログラムを入力させてているので、カスタマイズプログラムの入力を入力が容易なPC200で行うことができ、複合機100上でカスタマイズプログラムの作成編集を行う場合に比べて、操作表示部のカスタマイズをより容易に行うことができ、操作表示部のカスタマイズを効率的に行うことができる。

#### 【0122】

なお、実施の形態1にかかる複合機100では、インターネット220に接続されたクライアント端末としてのPC200の要求により、カスタマイズプログラムの入力処理を行っているが、インターネット以外のネットワーク、例えばLANなどに接続されたクライアント端末からの要求によりカスタマイズプログラムの入力処理を行うように構成することも可能である。また、通信プロトコルとしては、TCP/IP以外のプロトコルを利用しても良い。

#### 【0123】

(実施の形態2)

実施の形態1にかかる複合機100は、ネットワーク接続されたクライアント端末であるPC200で操作表示部のカスタマイズプログラムを作成して複合機100に保存するものであったが、この実施の形態2にかかる複合機は、カスタマイズプログラムの入力を複合機100のオペレーションパネルから行えるものである。

#### 【0124】

図12は、実施の形態2にかかる複合機1100における主要構成を示すブロック図である。なお、複合機1100の他の構成については、図2に示した実施の形態1の複合機100の構成と同様である。図12に示すように、本実施の形態にかかる複合機1100は、複合機初期化部129と、プログラム起動部131と、プログラミングサービス1132と、インタプリタ（シェル）134と、HDD103とを備えた構成となっている。

#### 【0125】

プログラミングサービス1132は、オペレーションパネル210の操作表示部にプログラミング画面を表示し、表示されたプログラミング画面からオペレーションパネル210の操作表示部のカスタマイズプログラムを入力させるものである。また、プログラミングサービス1132は、入力されたカスタマイズプログラムをHDD103のカスタマイズ領域として作成されたカスタマイズディレクトリ212に保存する。さらに、プログラミングサービス1132は、複合機1100上でカスタマイズプログラムが実行されるように、起動設定ファイルに入力されたカスタマイズプログラムを登録する。

#### 【0126】

なお、複合機初期化部129、プログラム起動部131、シェル134の機能およびHDD103の起動設定ファイル211およびカスタマイズ領域としてのカスタマイズディレクトリについては、実施の形態1にかかる複合機100と同様である。

#### 【0127】

図13は、プログラミング画面1101の一例を示す説明図である。本実施の形態にかかる複合機1100において、オペレーションパネル210の操作表示部のカスタマイズを行うためには、システム初期設定画面からプログラミングのタグを選択してプログラミング画面を表示させる。

#### 【0128】

このプログラミング画面1101は、実施の形態1における同画面とほぼ同様に、入力フィールドと、参照ボタンと、保存ボタンと、実行設定ボタンと、テスト実行ボタンとが表示されている。かかるプログラミング画面1101では、タ

ツチ操作でキー入力を行うソフトウェアキーボードを操作表示部に表示させて、入力フィールドにカスタマイズプログラムを入力することが考えられる。

### 【0129】

しかしながら、操作表示部の領域サイズや入力の困難さから直接カスタマイズプログラムを入力することは困難である。このため、予めHDD103などの記憶媒体の所望の領域に、PCなどで作成したカスタマイズプログラムをネットワーク経由で格納し、プログラミング画面1101上の参照ボタンを押下して、カスタマイズプログラムを選択することにより、入力フィールドにカスタマイズプログラムを入力させる。ソフトウェアキーボードを表示させ、プログラムの簡単な修正などだけを入力フィールドで行うという開発作業が好ましい。

### 【0130】

そして、カスタマイズプログラムが完成したら、保存ボタンを押下する。これによって、実施の形態1と同様に、プログラミングサービス1132によって、カスタマイズプログラムをHDD103のカスタマイズディレクトリに保存される。また、実行設定ボタンを押下すると、実施の形態1と同様に、プログラミングサービス1132によって、カスタマイズプログラムとカスタマイズプログラムで指定されたボタン（キー）のキーコードが対応付けられてテーブルとして保存される。なお、作成したカスタマイズプログラムの実行処理については、実施の形態1の複合機100と同様である。

### 【0131】

このように実施の形態2にかかる複合機1100では、プログラミングサービス1132によって、オペレーションパネル210の操作表示部にプログラミング画面1101を表示し、この画面からカスタマイズプログラムを入力させてないので、ネットワークに接続されていないスタンドアローンの複合機1100においても操作表示部のカスタマイズを行うことができる。

#### （実施の形態3）

実施の形態1および2の複合機100、1100は、カスタマイズプログラムと操作表示部のボタン（キー）との対応付けをカスタマイズプログラムで行うものであったが、この実施の形態3にかかる複合機は、複合機の電源投入時にカス

タマイズプログラムと操作表示部のボタン（キー）との対応付けを動的に行うものである。

#### 【0132】

図14は、実施の形態3にかかる複合機1300の機能的構成を示すブロック図である。本実施の形態の複合機1300は、実施の形態1と同様に、ネットワーク上のPC200からシェルスクリプトのカスタマイズプログラムの作成を行って、複合機1300のHD103に作成されたカスタマイズプログラムを保存して実行するようになっている。

#### 【0133】

図14に示すように、本実施の形態の複合機1300は、オペパネ登録部133を備えている点が図2に示す実施の形態1の複合機100と異なっており、他の構成は実施の形態1の複合機100と同様である。

#### 【0134】

オペパネ登録部133は、カスタマイズプログラムを実行させるためのボタン（キー）の割り付けを行うものである。すなわち、オペパネ登録部133によって割り付けたボタン（キー）をオペレーションパネル上で押下（タッチ操作）することによって、カスタマイズプログラムが実行されるようになっている。

#### 【0135】

PC200上からのカスタマイズプログラムの作成および保存の処理は、実施の形態1のプログラミングサービス131と同様に行われる。このため、カスタマイズプログラムと操作表示部に表示されるボタン（キー）の割付処理について説明する。カスタマイズプログラムのカスタマイズディレクトリへの保存が完了し、作成したカスタマイズプログラムをオペレーションパネル210のボタン（キー）に割り付けるため、利用者は複合機100の電源を再投入する。

#### 【0136】

図15は、複合機1300の電源投入が行われてから、カスタマイズプログラムとボタン（キー）との割り付けが完了するまでの一連の処理の手順を示すフローチャートである。複合機1300の電源投入からプログラム起動部131までの処理（ステップS1501～S1506）については、実施の形態1の複合機

100による処理（ステップS1001～S1006）と同様である。

#### 【0137】

起動されたプログラム起動部131は、まずHDD103の起動設定ファイル211を参照し（ステップS1507）、起動すべきプログラムの名称を取得する。図16は、本実施の形態で使用される起動設定ファイル211の内容の一例を示す説明図である。この例では、オペパネ登録部のプログラム、シェルおよび外部アプリであるXXXアプリを順に起動するように設定されている。

#### 【0138】

このため、プログラム起動部131は、まずオペパネ登録部133のプログラムを起動する（ステップS1508）。なお、このオペパネ登録部133のプログラムは、予めカスタマイズディレクトリに格納されている。次いで、インタプリタ（シェル）134を起動し（ステップS1509）、最後に、登録されている外部アプリを起動する（ステップS1510）。

#### 【0139】

起動されたオペパネ登録部133は、キー割り付け設定画面をオペレーションパネルに表示する（ステップS1511）。図17は、キー割り付け設定画面の一例を示す説明図である。図17に示すように、キー割り付け設定画面には、各カスタマイズプログラムごとに割り付け可能なボタン（キー）が表示され、かかる画面で所望のボタン（キー）を押下（タッチ操作）すると、オペパネ登録部133は、該当するカスタマイズプログラムの名称が押下されたボタン（キー）のキーコードと対応付けてHDD103またはフラッシュメモリなどの記憶媒体にテーブル情報として格納する（ステップS1512）。これによって、オペレーションパネルに表示されるボタン（キー）にカスタマイズプログラムが割り付けられることになる。

#### 【0140】

このように実施の形態3にかかる複合機1300では、プログラミングサービス132によって、PC200に対し操作表示部からのキー操作入力に対する実行処理をシェルで記述したカスタマイズプログラムのプログラミング画面201を表示してカスタマイズプログラムを入力させ、オペパネ登録部133によって

、カスタマイズプログラムを操作表示部に表示および入力を行うボタン（キー）に割り付けて、このオペパネ登録部133を複合機1300の電源投入時に起動しているので、カスタマイズプログラムが実行される場合のボタン操作を複合機1300の起動時に動的に定めることができ、操作表示部のカスタマイズを柔軟的に行うことができる。

#### 【0141】

なお、本実施の形態では、オペパネ登録部133によってボタン（キー）への割り付け処理を行っているが、カスタマイズプログラムより先に起動されるプログラムでかかる割り付け処理を行うように構成してもよい。例えば、SCS122などのコントロールサービスでボタン（キー）割り付け処理を行えば、割り付け処理のために別途プログラムを起動する必要が無くなり、効率的な処理が行えるという利点がある。

#### （実施の形態4）

実施の形態1～3の複合機100、1100、1300は、ボタン押下による実行処理をシェルスクリプトで作成したカスタマイズプログラムを複合機100、1300の汎用OS121で用意されたコマンドインタプリタであるシェル134によって実行するものであったが、この実施の形態4にかかる複合機は、汎用OS121の配下で動作するインタプリタを使用してカスタマイズプログラムを実行するものである。

#### 【0142】

図18は、実施の形態4にかかる複合機1700を備えたネットワーク構成図であり、図19は、実施の形態4にかかる複合機1700の機能的構成を示すブロック図である。実施の形態4にかかる複合機1700は、実施の形態1と同様に、TCP/IPを利用したインターネットに接続され、当該ネットワークに接続されたクライアント端末であるPC200から複合機の操作を行うオペレーションパネルの操作表示部のカスタマイズプログラムの入力およびデバッグ作業を行うことを可能としたものである。

#### 【0143】

ただし、本実施の形態の複合機1700では、オペレーションパネル210の

操作表示部に表示されるボタン（キー）を押下したときの実行処理を、Visual Basic（登録商標）言語でカスタマイズプログラムを記述する。そして、作成されたカスタマイズプログラムを、汎用OS121の配下で動作するVisual Basic インタプリタで実行させている。

#### 【0144】

図18に示すように、複合機100は、複合機初期化部129と、プログラム起動部131と、プログラミングサービス132と、Visual Basic インタプリタ1734（以下、「VBインタプリタ1734」という。）と、NCS128と、httpd106と、共有メモリ105と、HDD103とを備えた構成となっている。なお、プログラミングサービス132、複合機初期化部129、プログラム起動部131、NCS128、httpd106の構成および機能、HDD103に格納される起動設定ファイル211の内容およびカスタマイズディレクトリについては、実施の形態1の複合機100と同様である。

#### 【0145】

VBインタプリタ1734は、カスタマイズプログラムをステップごとに解析し逐次実行するものである。このVBインタプリタ1734は、汎用OS121の配下で実行されるものである。本実施の形態において、カスタマイズプログラムの作成および保存は、実施の形態1のプログラミングサービス132と同様に行われる。

#### 【0146】

また、本実施の形態の複合機1700では、VBインタプリタ1734によってカスタマイズプログラムを実行させるため、複合機1700の起動時にVBインタプリタ1734も起動させておく必要がある。このため、実施の形態1と同様の起動設定ファイル211にVBインタプリタ1734のプログラム名称を予め設定しておく必要がある。

#### 【0147】

図20は、起動設定ファイルの内容の一例を示す説明図である。図20に示すように、本実施の形態の複合機1700で使用する起動設定ファイル211には、先頭の行に「VBインタプリタプログラム」が設定されている。このため、P

ログラム起動部 131 は、起動設定ファイル 211 を参照した後、まず VB インタプリタ 1734 を起動することになり、これによってカスタマイズプログラムの実行が可能となる。

#### 【0148】

このように実施の形態 4 にかかる複合機 1700 では、汎用 OS 121 の配下で動作する VB インタプリタによってカスタマイズプログラムを実行するので、汎用 OS で提供されるインタプリタよりも高度な処理を行えるインタプリタを利用することができ、操作表示部のカスタマイズの範囲を拡張することができる。

#### 【0149】

なお、実施の形態 4 では、PC 200 上では VB インタプリタはインストールされていないが、PC 200 に Visual Basic 開発環境をインストールし、操作表示部の画面設計を PC 200 上で行って、生成された画面ファイルをファイル転送などをを利用して、複合機 1700 のカスタマイズディレクトリに格納するように構成しても良い。

#### 【0150】

また、実施の形態 4 の複合機 1700 では、インタプリタとして VB インタプリタ 1734 を使用し、開発するカスタマイズプログラムを Visual Basic 言語で記述していたが、この他、インタプリタ用の言語を用いても良い。例えば、複合機 1700 に Java (登録商標) VM (仮想マシン)、Java (登録商標) インタプリタを搭載し、カスタマイズプログラムを Java (登録商標) 言語で記述するように構成することもできる。

#### 【0151】

実施の形態 1 ~ 4 にかかる複合機では、プログラミングサービス 132 をコントロールサービス層に搭載した構成としているが、アプリケーション層に搭載した構成とすることもできる。

#### 【0152】

(実施の形態 5)

実施の形態 4 にかかる複合機は、汎用 OS 121 の配下で動作するインタプリタを使用してカスタマイズプログラムを実行するものであったが、この実施の形

態5にかかる複合機1800は、VBインタプリタ1801とVBアプリ1802をアプリケーション層に搭載した構成としたものである。

#### 【0153】

図21に、実施の形態5に係る複合機の機能構成のブロック図を示す。実施の形態5にかかる複合機は、実施の形態4と同様に、TCP/IPを利用したインターネットに接続され、当該ネットワークに接続されたクライアント端末であるPC200から、オペレーションパネルの操作表示部等のカスタマイズプログラムの入力およびデバッグ作業を行うことを可能としたものである。ただし、実施の形態5では、作成されたカスタマイズプログラム（以下、VBアプリと称する）を、アプリケーション層に搭載されたVisual Basic インタプリタ（以下、VBインタプリタと称する）で実行させている。なお、図中、プログラミングサービス132をアプリケーション層に搭載した構成としてもよい。

#### 【0154】

Visual Basic 開発環境をインストールしたPC200上で操作表示部等のVBアプリの作成を行い、その作成したVBアプリを複合機に転送し、複合機でそのVBアプリを格納しておき、オペレーションパネルにおけるアプリ選択等に応じてVBアプリを起動してVBインタプリタ1801により実行するようとする。

#### 【0155】

以下、より具体的に説明する。

#### 【0156】

図22は、PC200上でVisual Basic開発環境を用いてVBアプリを作成している場面の例を示す図である。

#### 【0157】

同図中の画面内にはProject 2-Form 2 (Form)の表題がついたウインドウと、Project 2-Form 2 (コード)の表題がついたウインドウが示されている。

#### 【0158】

Project 2-Form 2 (Form)の表題がついたウインドウは、複合機のオペレーションパネルで表示されるメニュー画面を作成するためのものであり、表示されてい

る状態では、3つのボタンが設定されており、一つは"SCAN"、一つは"FAX"あと一つは"START"のボタンである。これらのボタンが押された場合に、複合機のスキャニングやファックス送信などの動作が行われるように設計される。

#### 【0159】

Project 2-Form 2(コード)の表題がついたウインドウは、上記のパネルの画面中にあるボタンが押されたときの動作を記述するプログラミングを行うためのものである。

#### 【0160】

図22に示す状態でプログラムの保存を行うと、Project 2.vbpとForm 2.frmという二つのファイルが生成される。Project 2.vbpはプロジェクトを管理するための情報や、プログラムを実行する際の条件となるものが、変数への値の設定として保存されている。

#### 【0161】

Form 2.frmは、VB インタプリタ 1801によりインタプリトされるプログラムそのものであり、このVB アプリをPC 200から複合機上のプログラミングサービス 132を介してアップロードする。

#### 【0162】

そして、プログラミング画面 201でテスト実行ボタンが押下された場合には、複合機上のVB インタプリタ 1801が起動され、入力されたVB アプリの実行が行われる。このとき、テスト実行によって、オペレーションパネル 210に表示される画面と同一の画面をPC 200で動作しているブラウザに表示する。

#### 【0163】

図23、24に、VB インタプリタ 1801により実行されるVB アプリの例を示す。

#### 【0164】

図23に示すVB プログラムが複合機において実行されると、まず、NEXT WINボタンを含むウインドウが表示される。そして、NEXTWINボタンが押されると、call("change\_display", "Form2")により、関数change\_display("Form2")が実行され、次の画面 (Form2 ウィンドウ) に切り替わる。

**【0165】**

図24に、そのForm2 ウィンドウを表示するVBプログラムを示す。なお、Form2 のVBプログラムは、図22に示した開発状況を示す画面に対応する。

**【0166】**

Form2 ウィンドウでは、SCANボタンが押されたときに、

```
call("ScanStart", A4, ADF, TIFF, BINARY, "/work/tmpfile.tif")
gwOpItemCreate(win, MESSAGE_ITEM,
    ITEM_MESSAGE_X, 0,
    ITEM_MESSAGE_Y, 12,
    ITEM_MESSAGE_WIDTH, 100,
    ITEM_MESSAGE_HEIGHT, 12,
    ITEM_MESSAGE_CENTER_X, 0,
    ITEM_MESSAGE_BLINK, 0,
    ITEM_MESSAGE, "文字列", 0,
    ITEM_MESSAGE_FONT, FONT_12, 0,
    0);
```

が実行される。

**【0167】**

最初のScanStart関数は、A4サイズの原稿をADFから読み込んで、2値画像としてTIFFフォーマットでHDD上のファイル"/work/tempfile"で保管することを意味している。この関数をVBインタプリタ1801が解釈して、複合機に動作を行わせることとなる。

次の行のgwOpItemCreate()は複合機におけるOCS126によりその機能が提供されている関数であり、現在表示されている画面に文字列を表示する機能を実行するものである。gwOpItemCreate()は、VBインタプリタ1801により、OCS126への関数コールとして実行される。

**【0168】**

このようにして作成されたVBアプリを複合機で実行させるには、図11に示した起動設定ファイルにVBインタプリタとVBアプリを設定し、例えば図10

に示した手順と同様にして、VB インタプリタと VB アプリを起動する。

### 【0169】

複合機での VB アプリ実行時、オペレーションパネルに表示操作を行うためには次のような処理を行う。まず、VB インタプリタ 1801 は、SCS122 からの画面描画準備完了イベントを待ち、上記のイベントを受け取ったら、初期表示するウインドウイメージを作成する（ルートウインドウ生成）。次に、VB アプリが、ルートウインドウ内の子ウインドウの生成や文字列表示の関数を実行することにより初期画面を準備する。そして、画面準備完了を示す関数を実行することにより、画面の準備ができたことを SCS122 に通知する。次に、オペレーションパネル上でアプリ選択ボタンが選択されると、SCS122 から操作部オーナー移行要求のイベントが発行される。このイベントを VB アプリが受け取ると、操作部オーナー移行可能であることを示す関数を実行し、この時点で上記の初期画面が表示される。この後、VB アプリの実質的な処理が VB インタプリタ 1801 により実行される。

### 【0170】

上記のように、本実施の形態によれば、VB インタプリタを用いることにより効率良くアプリを開発することが可能となる。

#### （実施の形態 6）

実施の形態 5 にかかる複合機は、VB インタプリタをアプリケーション層に搭載した構成としたものであったが、実施の形態 6 にかかる複合機は、Java（登録商標）実行環境をアプリケーション層に搭載した構成としたものである。

### 【0171】

図 25 に、実施の形態 6 に係る複合機 1900 の機能構成のブロック図を示す。同図に示すように、実施の形態 6 に係る複合機 1900 は、アプリケーション層に Java（登録商標）実行環境 1901 と Java（登録商標）アプリ 1902 を搭載した構成を有しており、オペレーションパネルの操作表示部を Java（登録商標）アプリケーションとして開発し、その Java（登録商標）アプリケーションを複合機にダウンロードして Java（登録商標）実行環境 1901 を用いて実行することにより、操作表示部のカスタマイズ等を実現することが

可能となる。

#### 【0172】

図26に、複合機に搭載されるJava（登録商標）実行環境1901の構成の一例を示す。図26に示すように、本実施の形態におけるJava（登録商標）実行環境は、クラスライブラリ1911、仮想マシン1912、ネイティブプログラムインターフェース1913、及びアプリケーション管理部1914を有する。

#### 【0173】

クラスライブラリ1911は、Java（登録商標）アプリが容易に複合機を操作するためのサービスを提供するために用いられるクラスライブラリであり、ネイティブプログラムインターフェース1913の機構を用いてAPIにアクセスする。このクラスライブラリにより、機種依存部分の区別、簡易操作機能の提供、種々のオブジェクト指向クラスの提供がなされる。本実施の形態におけるクラスライブラリは、例えば、操作パネルクラス、イベントクラス、複合機制御クラス、ネットワークトランザクションクラス、基本的なJava（登録商標）クラスライブラリ等から構成される。

#### 【0174】

仮想マシン1912は、Java（登録商標）のソースコードプログラムをいったん中間コード形式にコンパイルし、その結果として得られるバイト・コードを解釈・実行する機能を有し、ネイティブプログラムインターフェース1913は、仮想マシン1912で実行されるJava（登録商標）コードがC言語などの他のプログラミング言語で書かれたアプリケーションやライブラリと相互運用するための機能を有している。

#### 【0175】

また、アプリケーション管理部1914は、Java（登録商標）アプリを管理する機能を有しており、例えば、Java（登録商標）アプリのリスト表示、起動や強制終了などのJava（登録商標）アプリ実行管理、Java（登録商標）アプリのインストールやバージョンアップ、インストール済みのJava（登録商標）アプリの削除、アプリケーション登録のためのパスワード設定等を行

う機能を有している。

#### 【0176】

さて、P C 2 0 0 から J a v a （登録商標）アブリ 1 9 0 2 を複合機にダウンロードし、複合機上で実行する処理は、これまでに説明した実施の形態と同様の方法で行うことが可能であるが、アプリケーション管理部 1 9 1 4 の機能を用いることにより、以下のような手順でダウンロードから起動までの処理を行うことも可能である。

#### 【0177】

（ダウンロードから起動までの手順の例）

アプリケーション管理部 1 9 1 4 は、他のアプリケーションと同様にして複合機の起動時に起動され、アプリケーション起動キー等を押すことにより、オペレーションパネルにユーザーインターフェースを表示する。このとき、J a v a （登録商標）アブリが登録されていない場合は、アプリケーションロード画面を表示する。

#### 【0178】

ユーザーは、このアプリケーションロード画面から所定のW e b サイト又は、I C カードにアクセスし、J a v a （登録商標）アブリをインストールする。

#### 【0179】

インストールの処理においては、J a v a （登録商標）アブリをインストールする前に、当該W e b サーバ等に対してJ a v a （登録商標）アブリに関する情報の問い合わせを行い、インストール可能であるJ a v a （登録商標）アブリであるか否かを確認する。確認項目は、例えば、プログラムサイズ、バージョン確認、プログラム最終更新情報、使用するメモリワークサイズ、使用するストレージサイズ、あらかじめ使用するネットワークアドレスの確認、利用可能期間（利用回数）、アプリケーションプログラム名等である。

#### 【0180】

上記の事項を確認し、インストール可能であると判断した場合は、アプリケーション管理部 1 9 1 4 がJ a v a （登録商標）アプリケーションファイルを複合機にダウンロードする。

### 【0181】

そして、Java（登録商標）アプリケーションプログラムのダウンロードが終了すると、アプリケーション管理部1914は、アプリケーション名を取得し、アプリケーションリストに追加し、同時にJava（登録商標）アプリケーションプログラムを複合機のハードディスクに格納する。

### 【0182】

以上の処理が完了することにより、ダウンロードしたJava（登録商標）アプリを利用できる状態になり、ユーザーは、アプリケーション管理部1914が提供するJava（登録商標）アプリケーション起動画面から起動するJava（登録商標）アプリを選択することにより、仮想マシンを起動し、Java（登録商標）アプリを実行することができる。

### 【0183】

（Java（登録商標）アプリケーションの開発）

さて、本実施の形態におけるJava（登録商標）アプリは、Java（登録商標）開発環境をインストールしたPC200上で作成し、これまでに説明した実施の形態と同様にして複合機上でテスト実行等を行うことにより開発を行うことができる。

### 【0184】

更に、図27に示すAPIのエミュレーション機能を開発環境に搭載することにより、ターゲットである複合機を用いずに開発することも可能である。これは、Java（登録商標）では、前述した仮想マシンを使用する構成となっているため、実行環境をエミュレートすることが比較的容易であることによる。エミュレータを用いた開発が完了したら、Java（登録商標）バイトコードをネットワーク又はICカードを介して複合機にインストールすることにより、複合機を用いた評価を行うことが可能である。

### 【0185】

（Java（登録商標）アプリケーションの例）

次に、Java（登録商標）アプリケーションの例について説明する。ここでは、操作パネルクラスを使用したサンプルプログラムを使用してJava（登録

商標) アプリの例について説明する。図28に、操作パネルクラスの階層構成をし、図29に、サンプルプログラムを示す。また、図30に実行結果の画面を示す。

### 【0186】

図29に示すプログラムは、複合機のオペレーションパネルにユーザーインターフェースを実装するプログラムである。図30に示すように、このプログラムは起動直後に”Hello World”及び”Your Input Was...”のメッセージと”Get Input...”のボタンを表示する。ユーザーが”Get Input...”のボタンに触れると、パネル上にソフトキーボードを表示し、ソフトキーボードには、”Add your Input”のタイトルが表示され、ユーザーに任意のコード入力を促す。ユーザーが、ソフトキーボードから任意の文字列、例えば、“ABCD!”を入力した場合、パネル上の”Your Input Was...”のメッセージの下にユーザーが入力した文字列を表示、すなわち、“ABCD!”を表示する。

### 【0187】

以下、図29に示すサンプルプログラムの内容について、プログラム中に記述した注釈の番号に沿って説明する。

### 【0188】

① は、操作パネルクラスライブラリを使用するための命令である。②は、GWApp を継承することを示し、これにより複合機のアプリケーション雛型（抽象）を使用することが可能となる。また、複雑な初期設定や終了処理をユーザーが記述せずに済み、メッセージ受信などの処理もユーザーに対して隠蔽することができる。また、③により “Hello World” のメッセージを表示するオブジェクトを生成し、④により “Get Input...” のボタンオブジェクトを生成する。そして、⑤により “Your Input Was...” のメッセージを表示するオブジェクトを生成し、⑥によりソフトキーボードから入力した文字を表示するメッセージオブジェクトを生成する。

### 【0189】

上記のように、本実施の形態によれば、Java（登録商標）環境を用いることにより効率良くアプリを開発することが可能となる。

本発明は、上記の実施の形態に限定されることなく、特許請求の範囲内で種々変更・応用が可能である。

#### 【発明の効果】

以上説明したように、請求項1にかかる発明によれば、操作表示部のカスタマイズプログラムの開発において、表示手段の入力画面から追加、修正ができ、また追加修正による再コンパイル、再リンクなしに画像形成装置で実行でき、操作表示部のカスタマイズを容易に行うことができるという効果を奏する。

#### 【0190】

また、請求項2にかかる発明によれば、プログラム実行時に実行するカスタマイズプログラムを容易に特定することができ、操作表示部のカスタマイズおよびその検証を容易に行うことができるという効果を奏する。

#### 【0191】

また、請求項3にかかる発明によれば、キー操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができるという効果を奏する。

#### 【0192】

また、請求項4にかかる発明によれば、カスタマイズプログラムを実際に画像形成装置で動作させる前に、表示手段上で簡易な検証を行うことができ、操作表示部のカスタマイズを効率的に行うことができるという効果を奏する。

#### 【0193】

また、請求項5にかかる発明によれば、カスタマイズプログラムの入力を当該入力が容易なクライアント端末で行うことができ、操作表示部のカスタマイズを効率的に行うことができるという効果を奏する。

#### 【0194】

また、請求項6にかかる発明によれば、ネットワークに接続されていないスタンダローンの画像形成装置においても操作表示部のカスタマイズを行うことができるという効果を奏する。

#### 【0195】

また、請求項7にかかる発明によれば、プログラム実行手段の起動を画像形成

装置の起動処理の一環として自動的に行うことができ、操作表示部のカスタマイズの検証をより効率的に行うことができるという効果を奏する。

#### 【0196】

また、請求項8にかかる発明によれば、別途インタプリタを画像形成装置にインストールすることなく、カスタマイズプログラムの実行を行うことができ、操作表示部のカスタマイズを容易に行うことができるという効果を奏する。

#### 【0197】

また、請求項9にかかる発明によれば、汎用OSで提供されるインタプリタよりも高度な処理を行えるインタプリタを利用することができ、操作表示部のカスタマイズの範囲を拡張することができるという効果を奏する。

#### 【0198】

請求項10に記載の発明によれば、プログラム実行手段をアプリケーションとして動作するインタプリタとすることができます。

#### 【0199】

また、請求項11に記載の発明によれば、前記カスタマイズプログラムは、前記プログラム実行手段により、前記コントロールサービスに対する関数呼び出しを行うこととするので、種々のアプリケーションを既存の関数を用いて作成することが可能となる。

#### 【0200】

また、請求項12にかかる発明によれば、操作表示部に表示されるキーの操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができるという効果を奏する。

#### 【0201】

また、請求項13にかかる発明によれば、カスタマイズプログラムが実行される場合のキー操作を画像形成装置の起動時に動的に定めることができ、操作表示部のカスタマイズを柔軟的に行うことができるという効果を奏する。

#### 【0202】

また、請求項14にかかる発明によれば、キーとカスタマイズプログラムの動

的な割り付けを簡易なユーザインターフェースで実現することができるという効果を奏する。

#### 【0203】

また、請求項15にかかる発明によれば、操作表示部のカスタマイズプログラムの開発において、表示手段の入力画面から追加、修正ができ、また追加修正による再コンパイル、再リンクなしに画像形成装置で実行でき、操作表示部のカスタマイズを容易に行うことができるという効果を奏する。

#### 【0204】

また、請求項16にかかる発明によれば、プログラム実行時に実行するカスタマイズプログラムを容易に特定することができ、操作表示部のカスタマイズおよびその検証を容易に行うことができるという効果を奏する。

#### 【0205】

また、請求項17にかかる発明によれば、キー操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができるという効果を奏する。

#### 【0206】

また、請求項18にかかる発明によれば、カスタマイズプログラムを実際に画像形成装置で動作させる前に、表示手段上で簡易な検証を行うことができ、操作表示部のカスタマイズを効率的に行うことができるという効果を奏する。

#### 【0207】

また、請求項19にかかる発明によれば、カスタマイズプログラムの入力を当該入力が容易なクライアント端末で行うことができ、操作表示部のカスタマイズを効率的に行うことができるという効果を奏する。

#### 【0208】

また、請求項20にかかる発明によれば、ネットワークに接続されていないストアンドアローンの画像形成装置においても操作表示部のカスタマイズを行うことができるという効果を奏する。

#### 【0209】

また、請求項21にかかる発明によれば、プログラム実行手段の起動を画像形

成装置の起動処理の一環として自動的に行うことができ、操作表示部のカスタマイズの検証をより効率的に行うことができるという効果を奏する。

#### 【0210】

また、請求項22にかかる発明によれば、操作表示部に表示されるキーの操作が行われたときの実行処理をカスタマイズプログラムによって行うことができ、キー操作における操作表示部のカスタマイズを容易に行うことができるという効果を奏する。

#### 【0211】

また、請求項23にかかる発明によれば、カスタマイズプログラムが実行される場合のキー操作を画像形成装置の起動時に動的に定めることができ、操作表示部のカスタマイズを柔軟的に行うことができるという効果を奏する。

#### 【0212】

また、請求項24にかかる発明によれば、キーとカスタマイズプログラムの動的な割り付けを簡易なユーザインターフェースで実現することができるという効果を奏する。

#### 【図面の簡単な説明】

##### 【図1】

実施の形態1にかかる複合機の主要構成およびネットワーク構成を示すブロック図である。

##### 【図2】

実施の形態1の複合機の機能的構成を示すブロック図である。

##### 【図3】

リクエストメッセージを受信した複合機における処理の手順を示すフローチャートである。

##### 【図4】

debug.cgiの処理の手順を示すフローチャートである。

##### 【図5】

プログラミング画面において、保存ボタン、実行設定ボタン、テスト実行ボタンのいずれかのクリック操作が行われた場合のWEBアプリの処理の手順を示す

フローチャートである。

【図 6】

プログラミング画面の内容の一例を示す説明図である。

【図 7】

プログラム入力フィールドに入力されるカスタマイズプログラムの一例を示す説明図である。

【図 8】

カスタマイズディレクトリに格納されたカスタマイズプログラムの名称の一例を示す説明図である。

【図 9】

キー対応付けテーブルの内容の一例を示す説明図である。

【図 10】

複合機の電源投入が行われてから、カスタマイズプログラムの実行が可能となるまでの一連の処理の手順を示すフローチャートである。

【図 11】

起動設定ファイルの一例を示す説明図である。

【図 12】

実施の形態 2 にかかる複合機における主要構成を示すブロック図である。

【図 13】

プログラミング画面の一例を示す説明図である。

【図 14】

実施の形態 3 にかかる複合機の機能的構成を示すブロック図である。

【図 15】

複合機の電源投入が行われてから、カスタマイズプログラムとボタン（キー）との割り付けが完了するまでの一連の処理の手順を示すフローチャートである。

【図 16】

起動設定ファイルの内容の一例を示す説明図である。

【図 17】

キー割り付け設定画面の一例を示す説明図である。

**【図 18】**

実施の形態 4 にかかる複合機の主要構成およびネットワーク構成を示すブロック図である。

**【図 19】**

実施の形態 4 にかかる複合機の機能的構成を示すブロック図である。

**【図 20】**

起動設定ファイルの内容の一例を示す説明図である。

**【図 21】**

実施の形態 5 にかかる複合機の機能的構成を示すブロック図である。

**【図 22】**

PC200上でVisual Basic 開発環境を用いてVBアプリを作成している場面を示す図である。

**【図 23】**

VB インタプリタにより実行される VB アプリの例を示す図である。

**【図 24】**

VB インタプリタにより実行される VB アプリの例を示す図である。

**【図 25】**

実施の形態 6 にかかる複合機の機能的構成を示すブロック図である。

**【図 26】**

複合機に搭載される Java (登録商標) 実行環境の構成の例を示す図である

**【図 27】**

Java (登録商標) 開発環境を説明するための図である。

**【図 28】**

操作パネルクラスの階層構成を示す図である。

**【図 29】**

サンプルプログラムを示す図である。

**【図 30】**

サンプルプログラムの実行結果を示す図である。

**【符号の説明】**

- 100, 1100, 1300, 1700, 1800, 1900 複合機  
101 白黒ラインプリンタ  
102 カラーラインプリンタ  
103 ハードディスク装置 (HDD)  
104 ハードウェアリソース  
105 共有メモリ  
106 h t t p d  
110 ソフトウェア群  
111 プリンタアプリ  
112 コピーアプリ  
113 ファックスアプリ  
114 スキヤナアプリ  
115 ネットファイルアプリ  
116 工程検査アプリ  
117 外部アプリ  
120 プラットホーム  
121 汎用OS  
122 SCS  
123 SRM  
124 ECS  
125 MCS  
126 OCS  
127 FCS  
128 NCS  
129 複合機初期化部  
130 アプリケーション  
131 プログラム起動部  
132 , 1132 プログラミングサービス

133 オペパネ登録部

134 インタプリタ（シェル）

135 カーネル

200 PC201, 1101プログラミング画面

211 起動設定ファイル

212 カスタマイズディレクトリ

220 インターネット

1734, 1801 VBインタプリタ

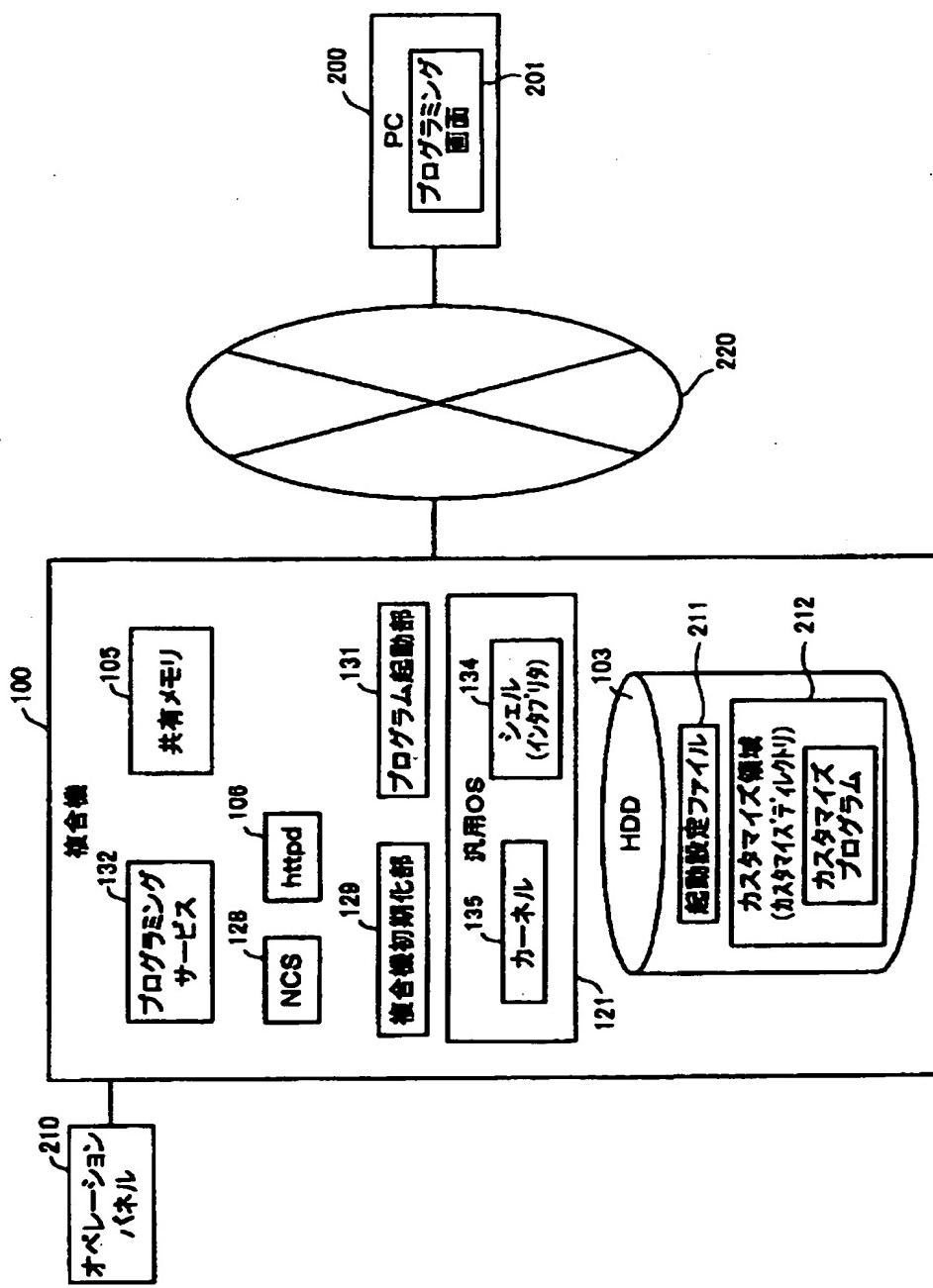
1901 Java（登録商標）実行環境

【書類名】

四面

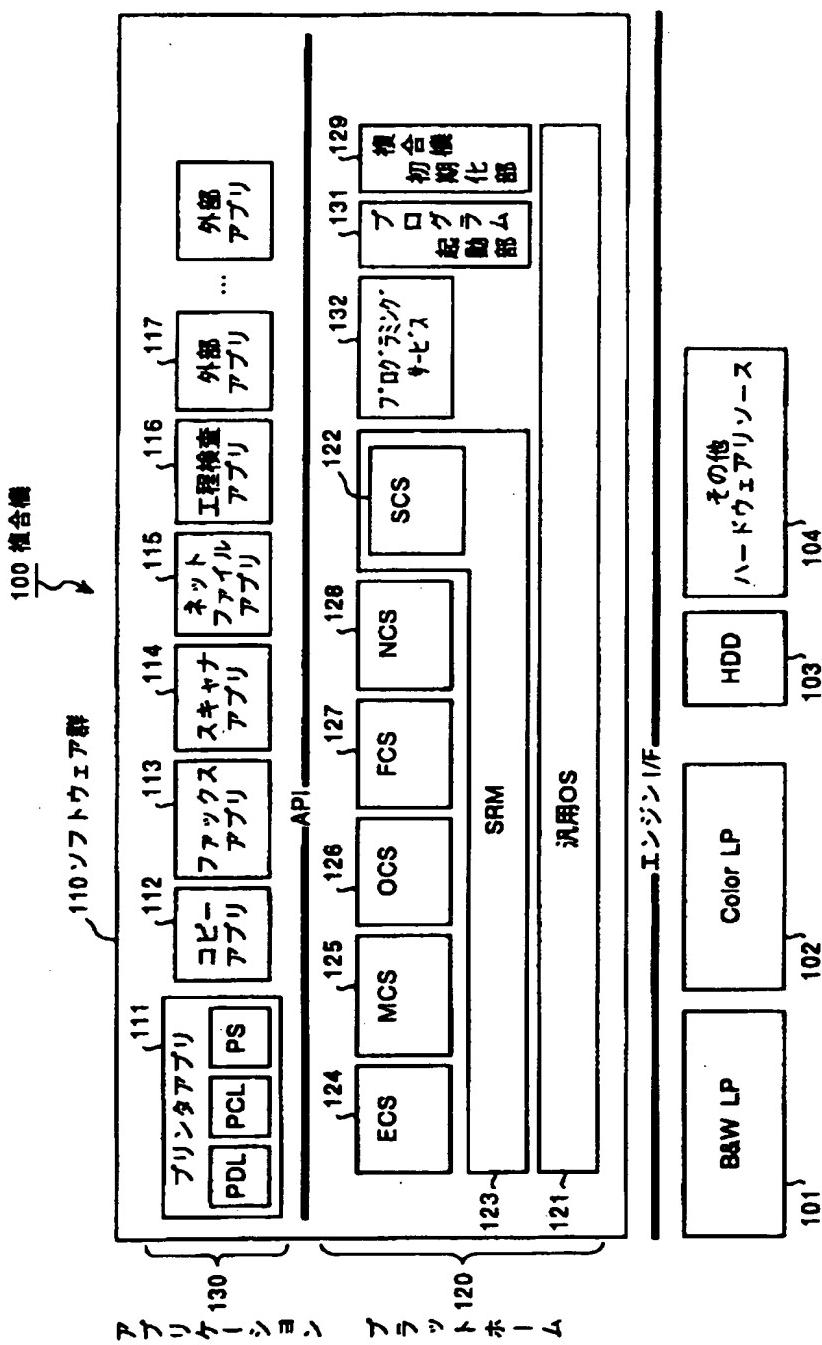
【図1】

## 実施の形態1にかかる複合機の主要構成および ネットワーク構成を示すブロック図



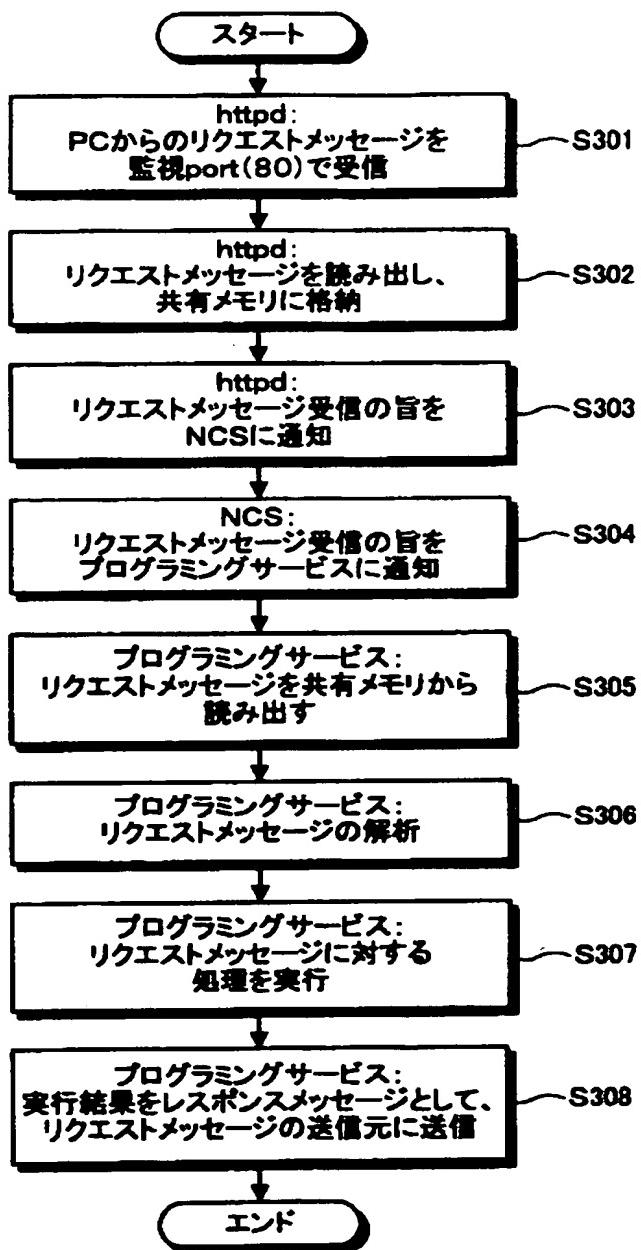
【図2】

## 実施の形態1の複合機の機能的構成を示すブロック図



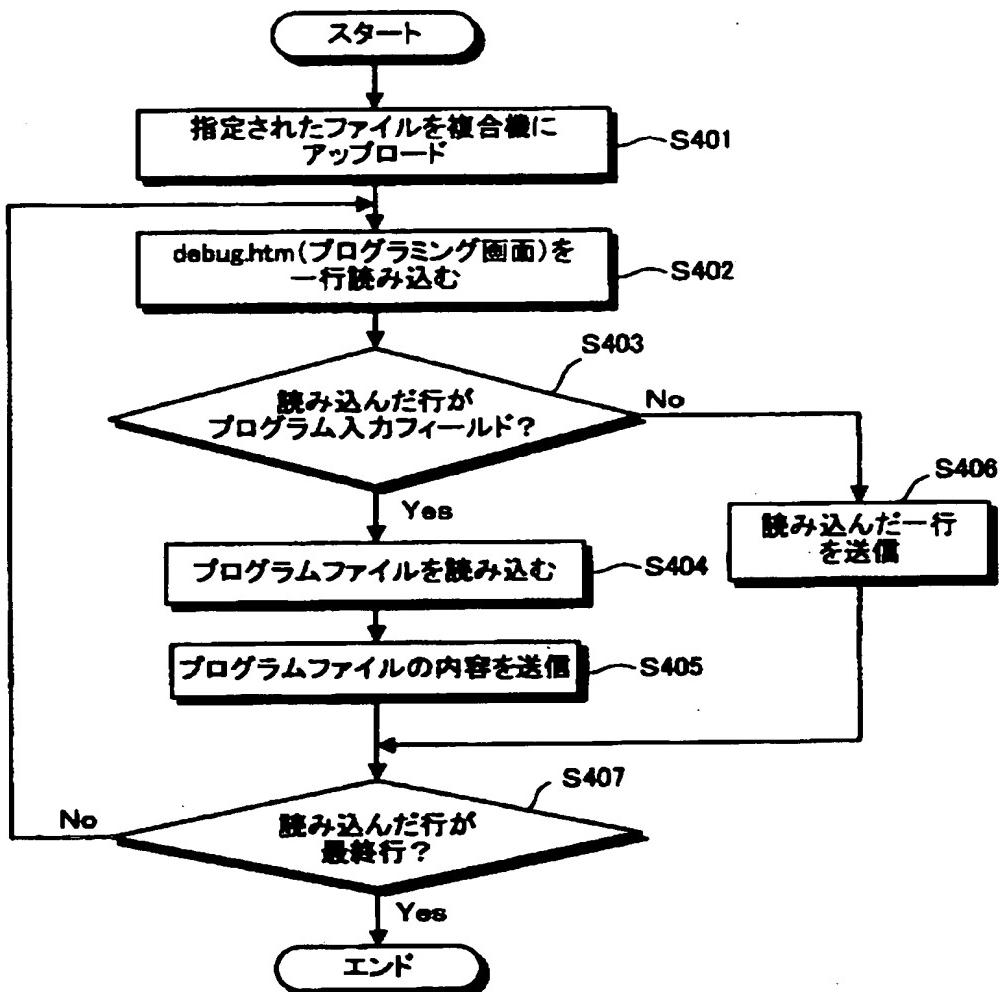
【図3】

リクエストメッセージを受信した複合機における  
処理の手順を示すフローチャート



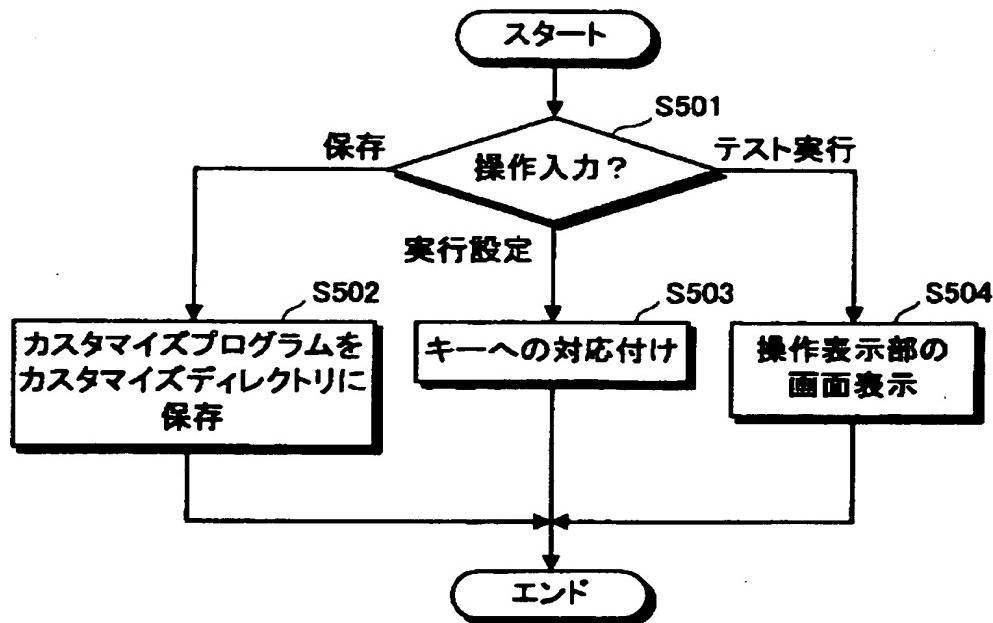
【図4】

## debug.cgiの処理の手順を示すフローチャート



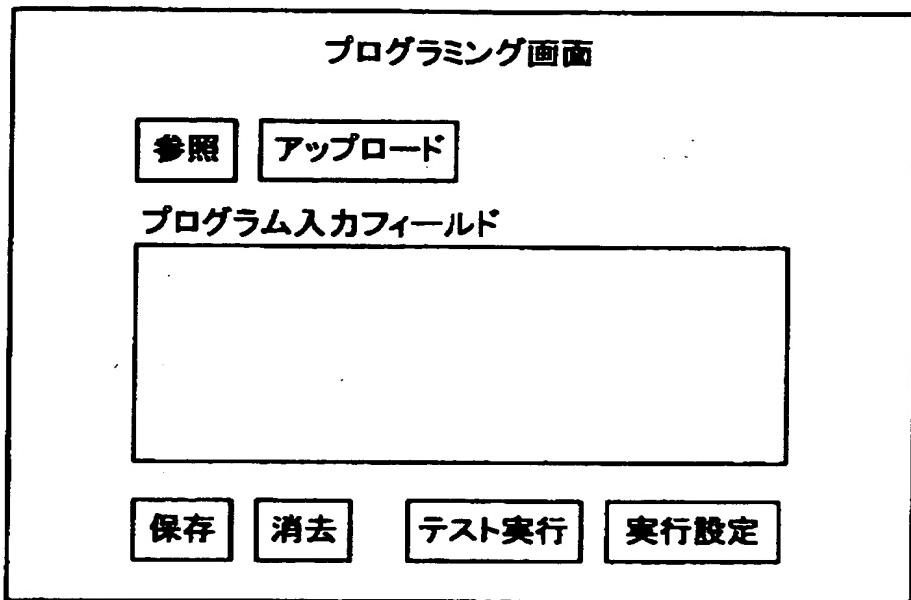
【図5】

プログラミング画面において、保存ボタン、実行設定ボタン、  
テスト実行ボタンのいずれかのクリック操作が行われた場合の  
WEBアプリの処理の手順を示すフローチャート



【図 6】

## プロムラミング画面の内容の一例を示す説明図



【図 7】

プロムラミング入力フィールドに入力される  
カスタマイズプログラムの一例を示す説明図

```
# ocr shell script
#
scainimage -out /hdd/ts/tmp/tmpimg.tif
decomp -in /hdd/ts/tmp/tmpimg.tif -out /hdd/ts/tmp/tmpimg.bmp
ocr -in /hdd/ts/tmp/tmpimg.bmp -out /hdd/ts/tmp/ocrresult.txt
mail xxxx@yy.zz.co.jp
```

【図8】

カスタマイズディレクトリに格納された  
カスタマイズプログラムの名称の一例を示す説明図

カスタマイズディレクトリ／ファイル名

hdd/xxx/opepane/shell1  
          shell2  
          shell3

【図9】

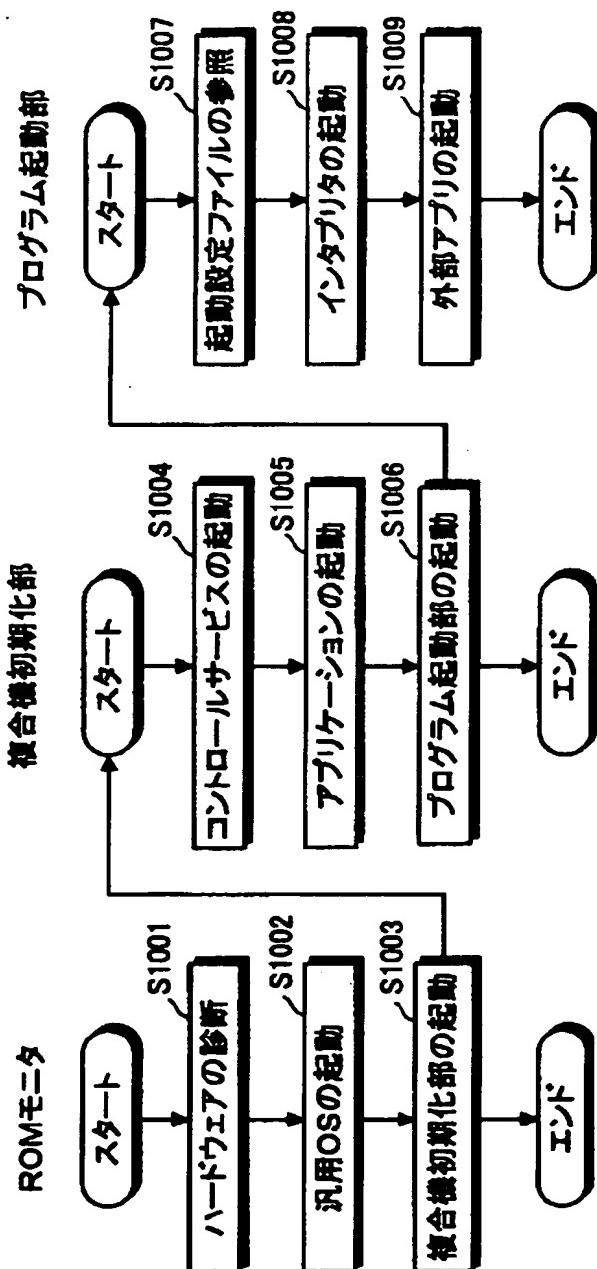
キー対応付けテーブルの内容の一例を示す説明図

キー対応付けテーブル

キーコード： カスタマイズプログラム  
501 : shell1  
502 : shell2  
503 : shell3

【図10】

複合機の電源投入が行われてから、カスタマイズプログラムの実行が可能となるまでの一連の処理の手順を示すフローチャート



【図 11】

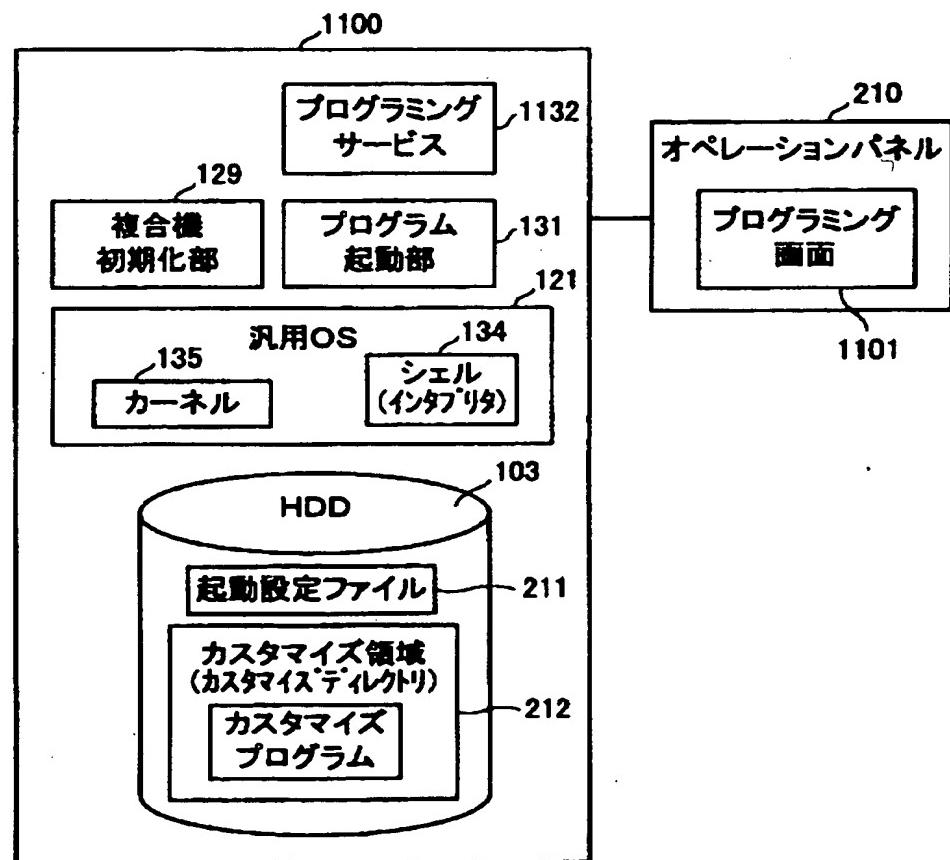
## 起動設定ファイルの一例を示す説明図

## 起動設定ファイル

プログラム名称	インタプリタ(シェル)
プログラム名称	XXXアプリ
	•
	•
	•

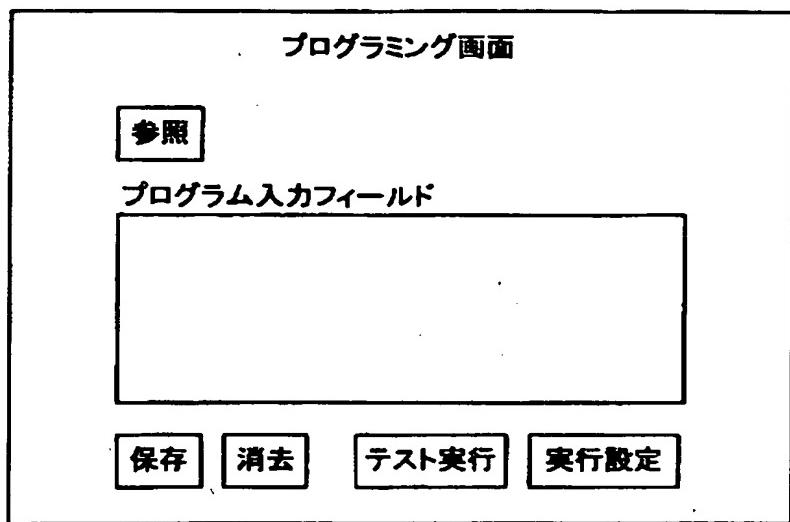
【図12】

実施の形態2にかかる複合機における主要構成を示すブロック図



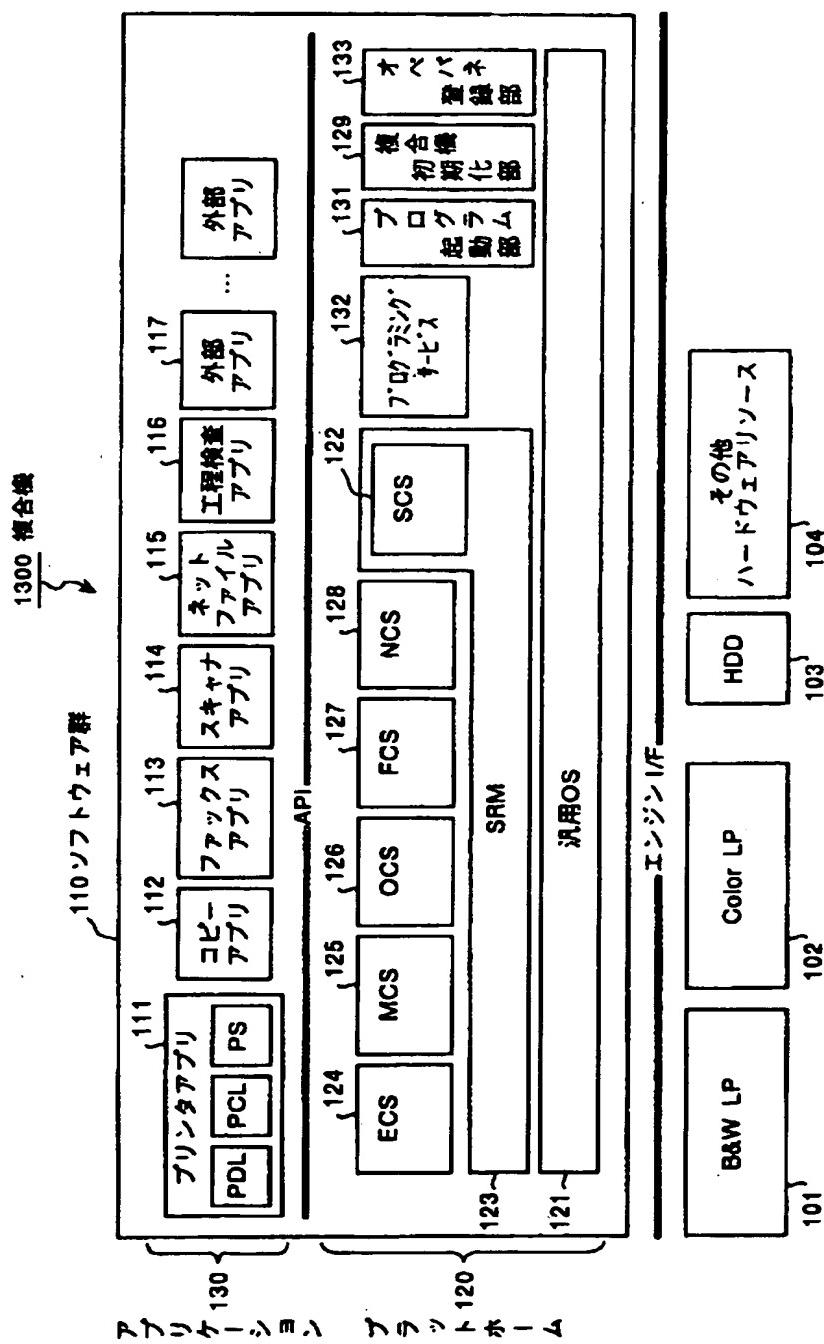
【図13】

## プロムラミング画面の一例を示す説明図



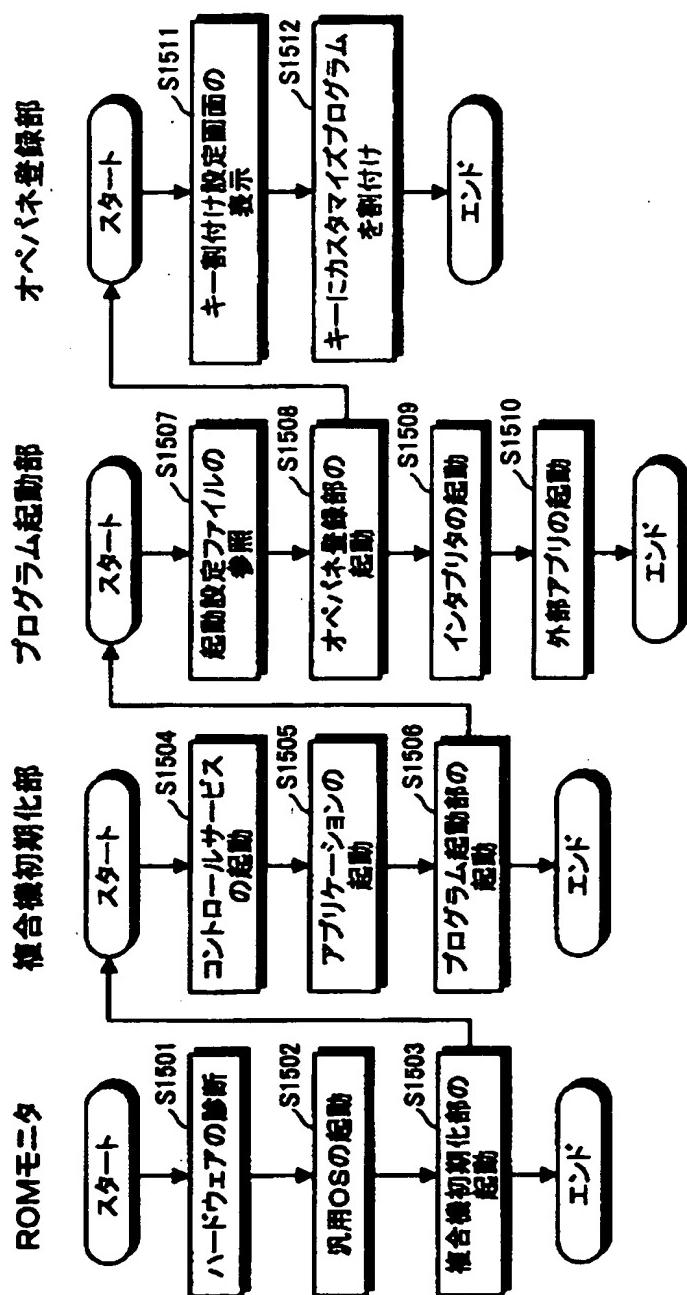
【図14】

実施の形態3にかかる複合機の機能的構成を示すブロック図



【図15】

複合機の電源投入が行われてから、カスタマイズプログラムとボタン(キー)との割り付けが完了するまでの一連の処理の手順を示すフローチャート



【図 16】

## 起動設定ファイルの内容の一例を示す説明図

## 起動設定ファイル

プログラム名称	オペバネ登録部プログラム
プログラム名称	インタプリタ(シェル)
プログラム名称	XXXアプリ
⋮	

【図 17】

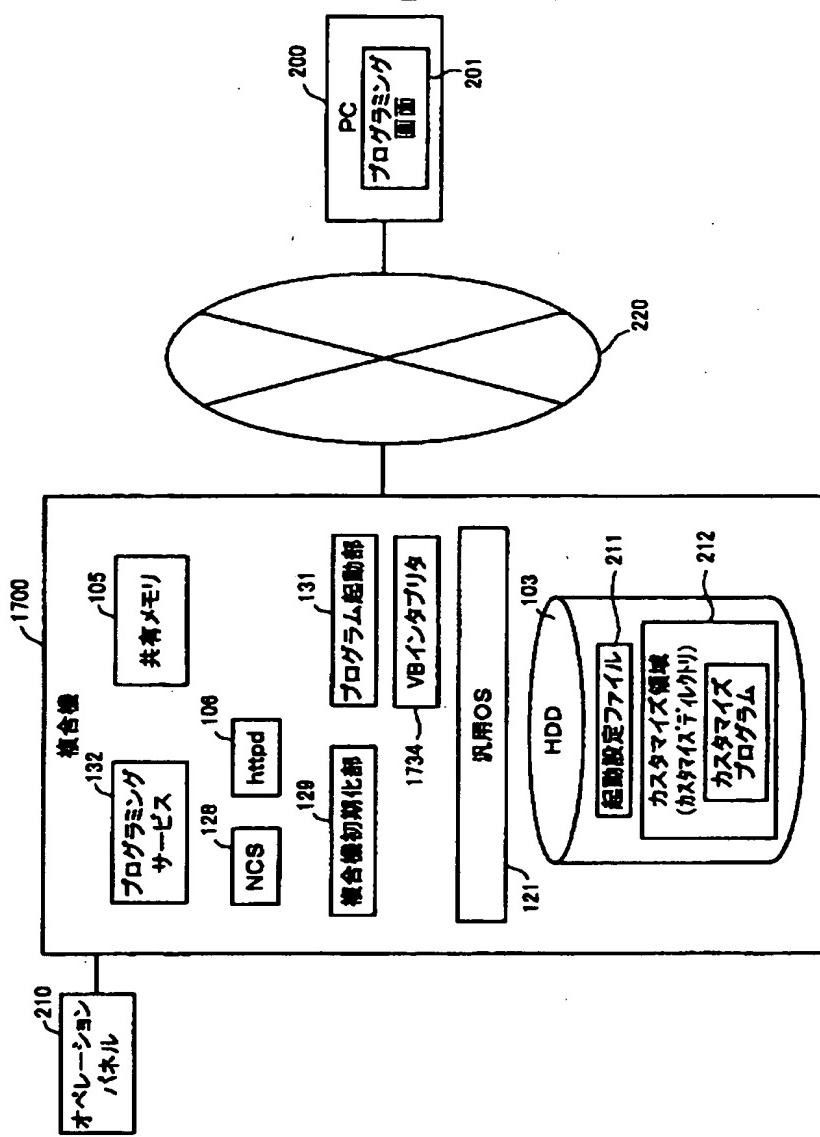
## キー割り付け設定画面の一例を示す説明図

**キー割付け設定画面**

カスタマイズプログラム	キー(ボタン)
Shell1	A      B      C
Shell2	A      B      C
Shell3	A      B      C

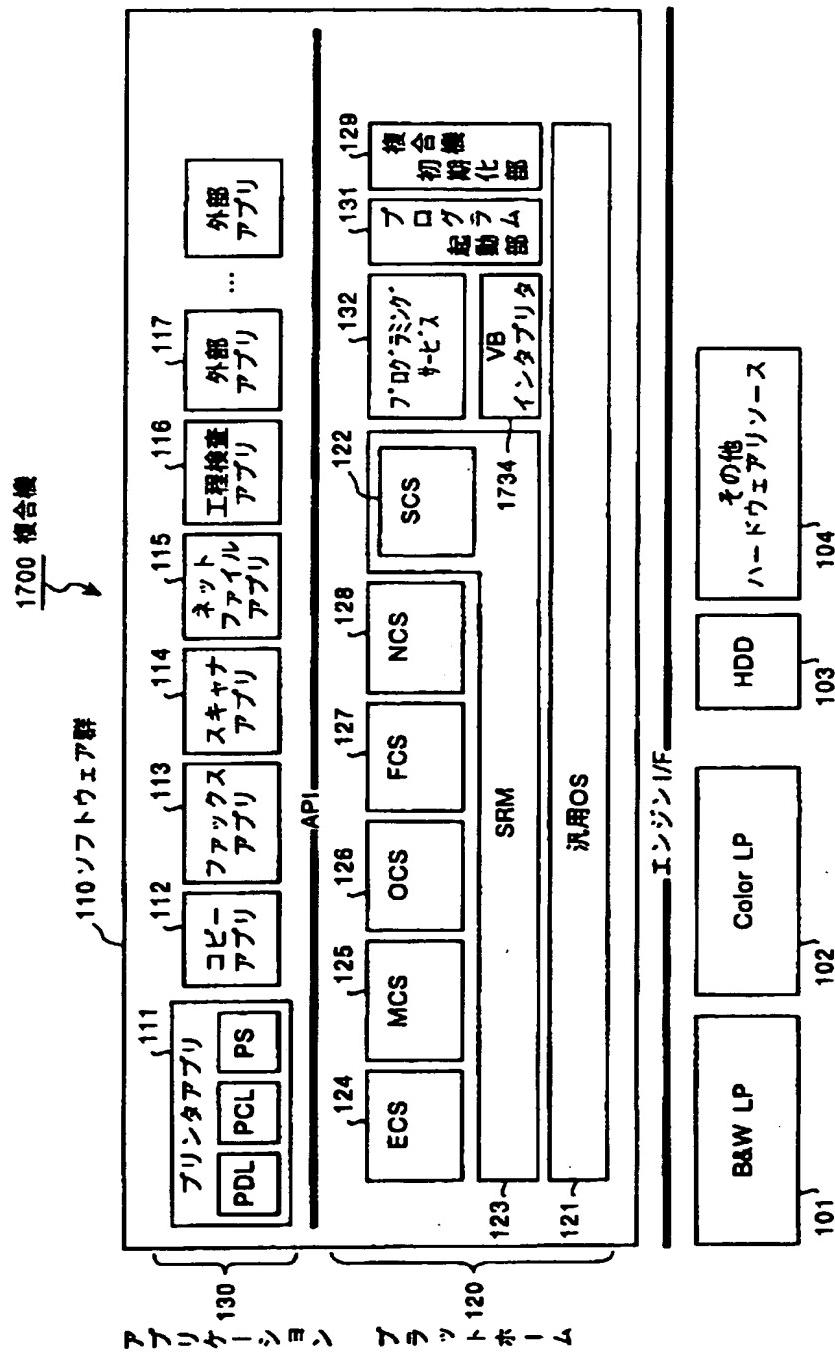
【図18】

実施の形態4にかかる複合機の主要構成および  
ネットワーク構成を示すブロック図



【図19】

実施の形態4にかかる複合機の機能的構成を示すブロック図



【図20】

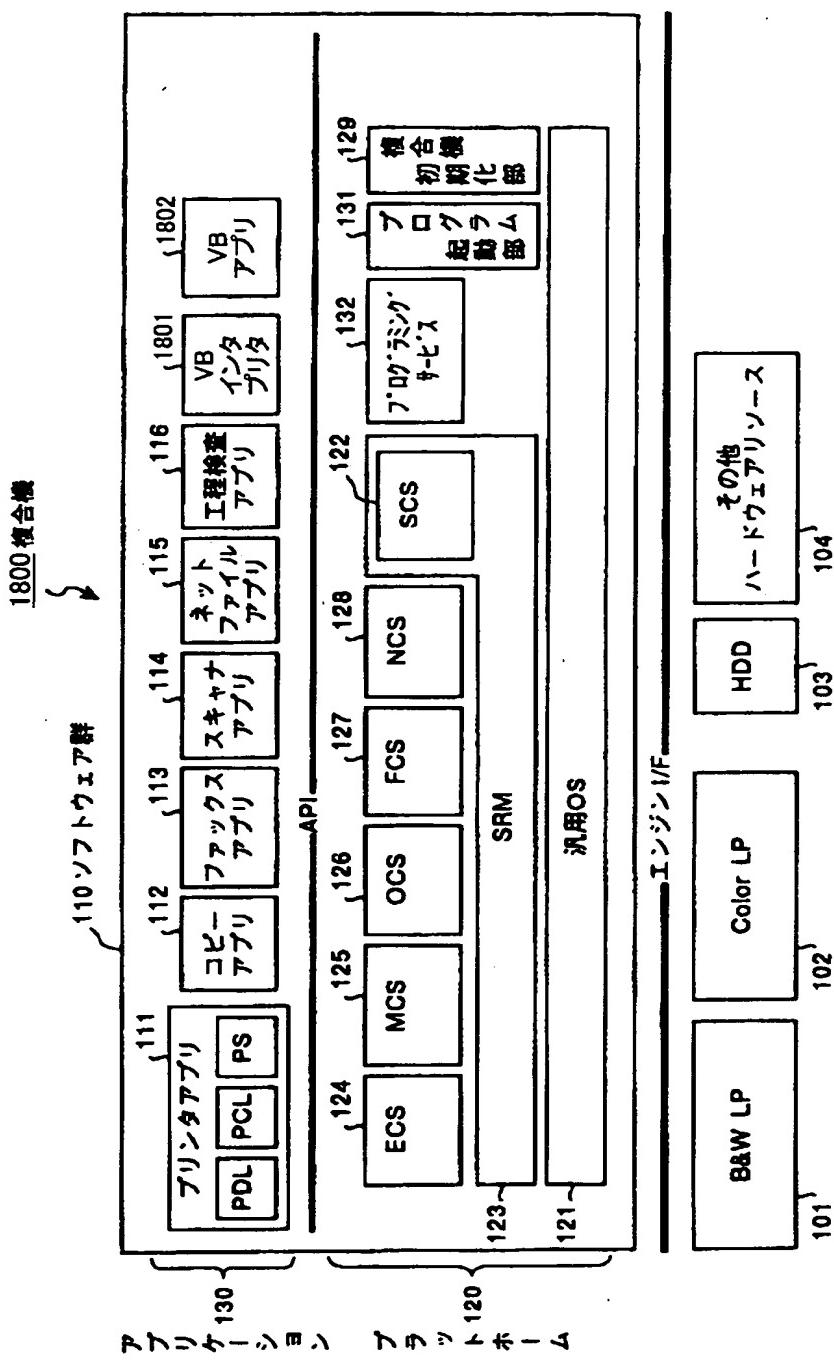
### 起動設定ファイルの内容の一例を示す説明図

#### 起動設定ファイル

プログラム名称	VBインタプリタ
プログラム名称	XXXアプリ
⋮	

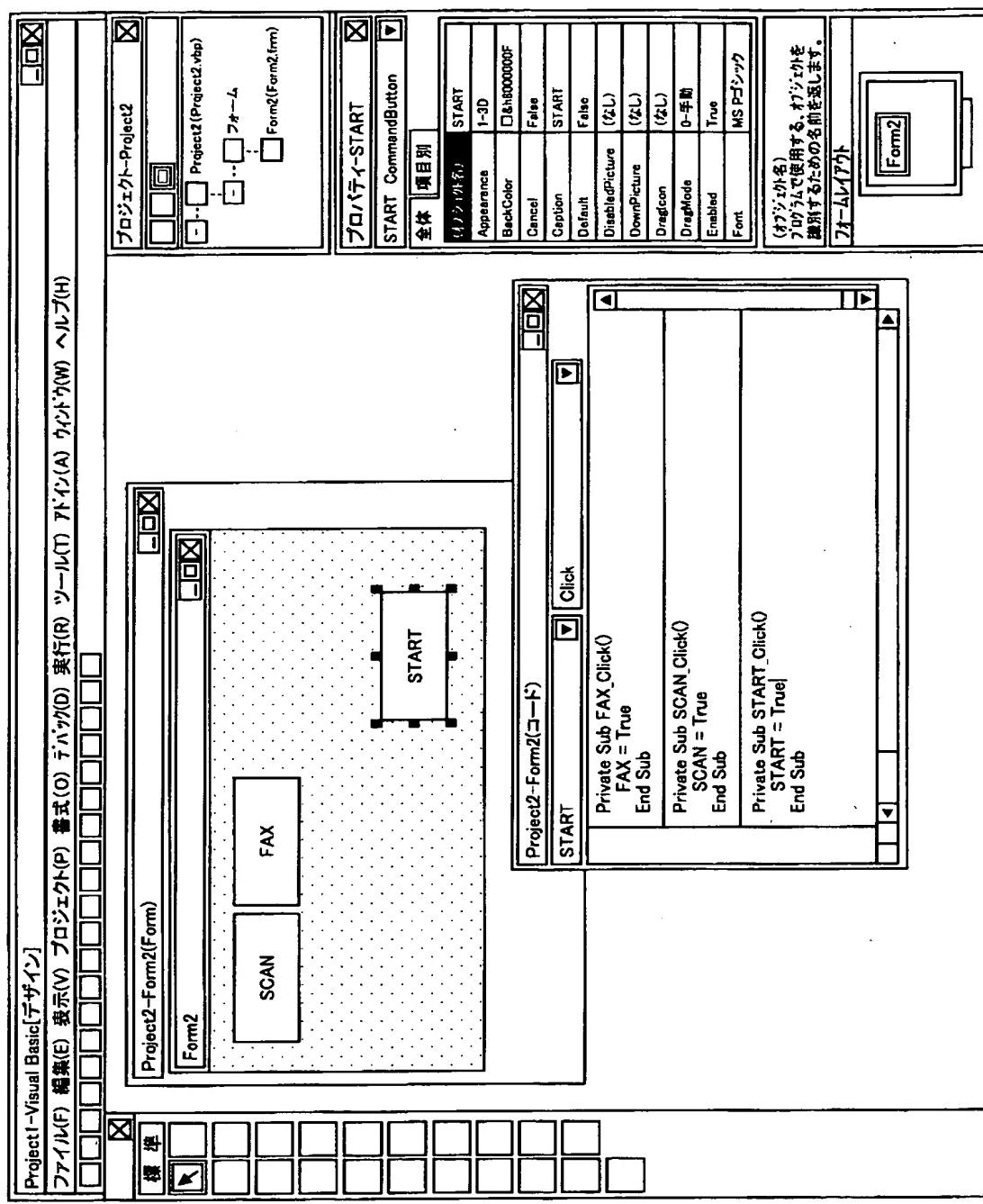
【図 21】

実施の形態5にかかる複合機の機能的構成を示すブロック図



【図22】

PC200上でVisual Basic開発環境を用いて  
VBアプリを作成している場面を示す図



## 【図23】

## VBインターフェリタにより実行されるVBアプリの例を示す図

```
Begin VB.Form Form1
    Caption           = "Form1"
    ClientHeight     = 3390
    ClientLeft       = 60
    ClientTop        = 345
    ClientWidth      = 6990
    LinkTopic        = "Form1"
    ScaleHeight      = 3390
    ScaleWight       = 6990
    StartUpPosition  = 3 'Windows の既定値
Begin VB.CommandButton NEXTWIN
    Caption          = "NEXTWIN"
    Height           = 735
    Left              = 480
    TabIndex          = 0
    Top               = 480
    Width             = 1575
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

Private Sub NEXTWIN_Click()
    /*Form2 ウィンドウへの切り替え */
    call("change_diplay", "From2")
End Sub
```

【図24】

## VBインターフェリタにより実行されるVBアプリの例を示す図

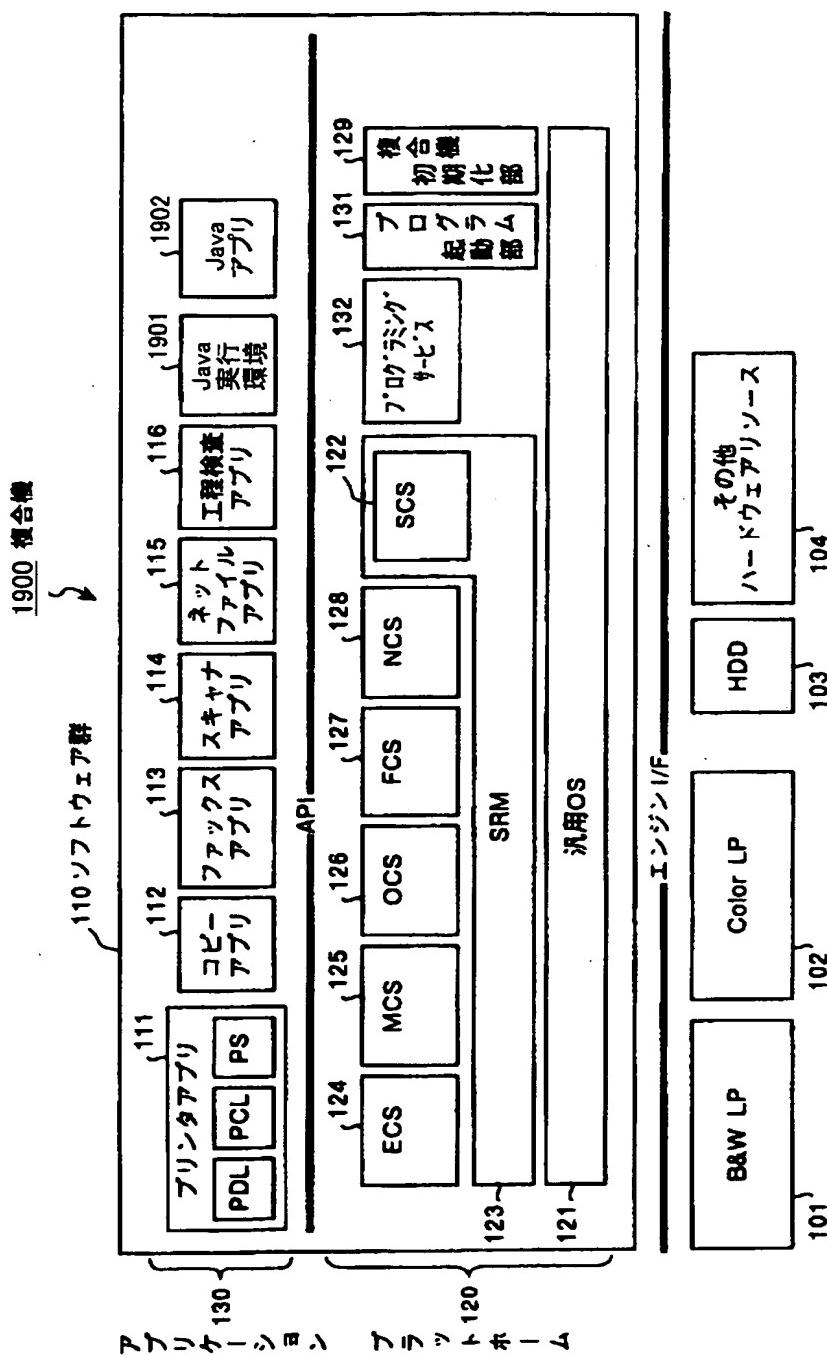
```

Begin VB.Form Form2
    Caption           =   "Form2"
    ClientHeight     =   3390
    ClientLeft       =   60
    ClientTop        =   345
    ClientWidth      =   6990
    LinkTopic        =   "Form2"
    ScaleHeight      =   3390
    ScaleWidth       =   6990
    StartUpPosition  =   3 'Windows の既定値
Begin VB.CommandButton START
    Caption          =   "START"
    Height           =   735
    Left              =   4560
    TabIndex         =   2
    Top               =   2040
    Width             =   1815
End
Begin VB.CommandButton FAX
    Caption          =   "FAX"
    Height           =   735
    Left              =   2520
    TabIndex         =   1
    Top               =   480
    Width             =   1455
End
Begin VB.CommandButton SCAN
    Caption          =   "SCAN"
    Height           =   735
    Left              =   480
    TabIndex         =   0
    Top               =   480
    Width             =   1575
End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Private Sub FAX_Click()
    FAX = True
End Sub
Private Sub SCAN_Click()
    call("ScanStart", A4, ADF, TIFF, BINARY, "/work/tmpfile.tif")
    gwOpItemCreate(win, MESSAGE_ITEM,
        ITEM_MESSAGE_X, 0,
        ITEM_MESSAGE_Y, 12,
        ITEM_MESSAGE_WIDTH, 100,
        ITEM_MESSAGE_HEIGHT, 12,
        ITEM_MESSAGE_CENTER_X, 0,
        ITEM_MESSAGE_BLINK, 0,
        ITEM_MESSAGE, "文字列", 0,
        ITEM_MESSAGE_FRONT, FONT_12, 0,
        0);
End Sub
Private Sub START_Click()
    START = True
End Sub

```

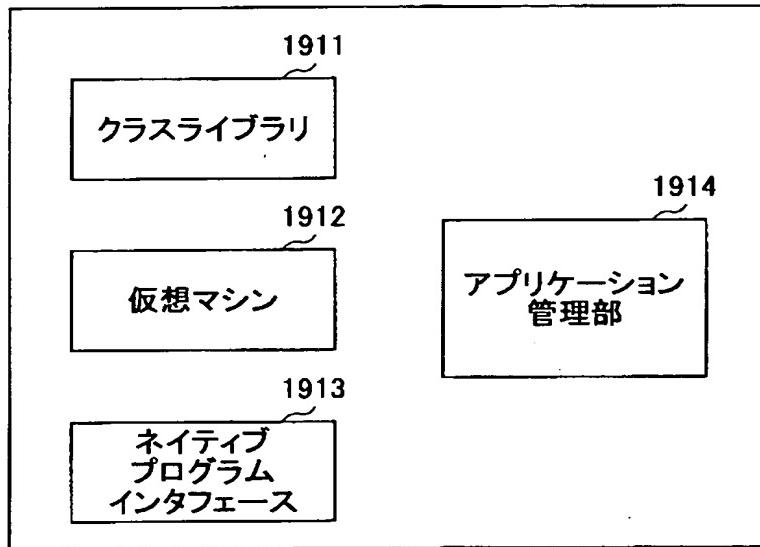
【図25】

実施の形態6にかかる複合機の機能的構成を示すブロック図



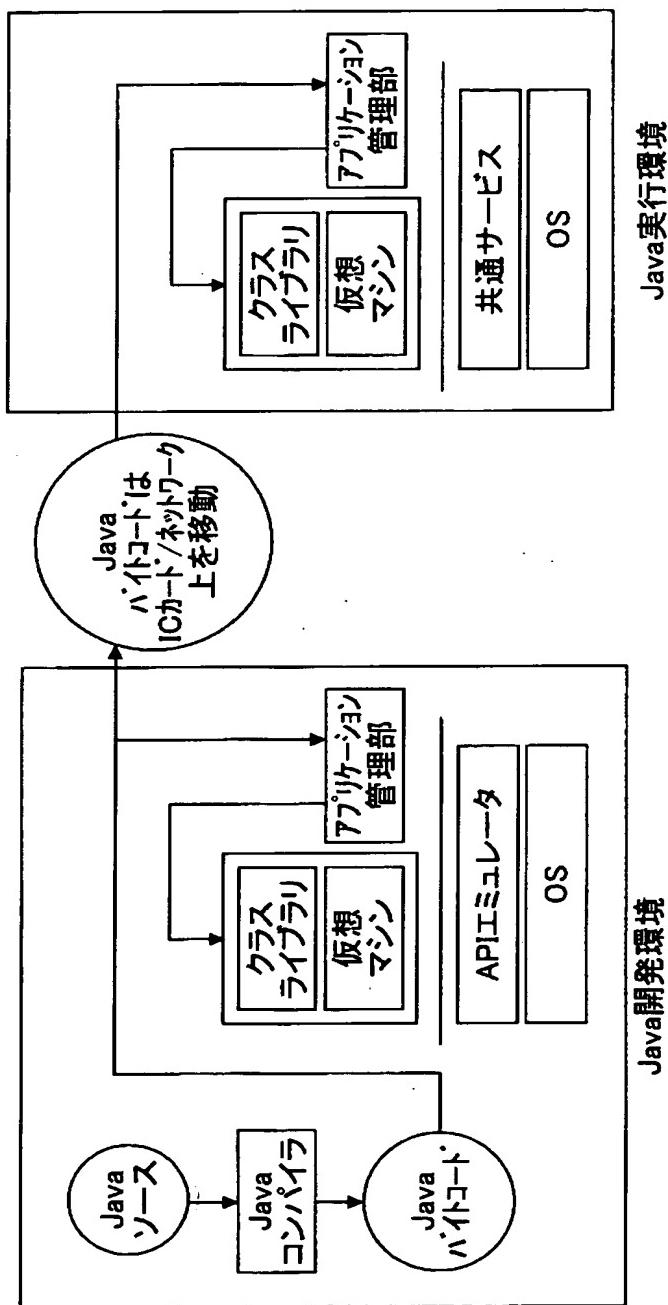
【図 26】

複合機に搭載されるJava(登録商標)実行環境の構成の例を示す図



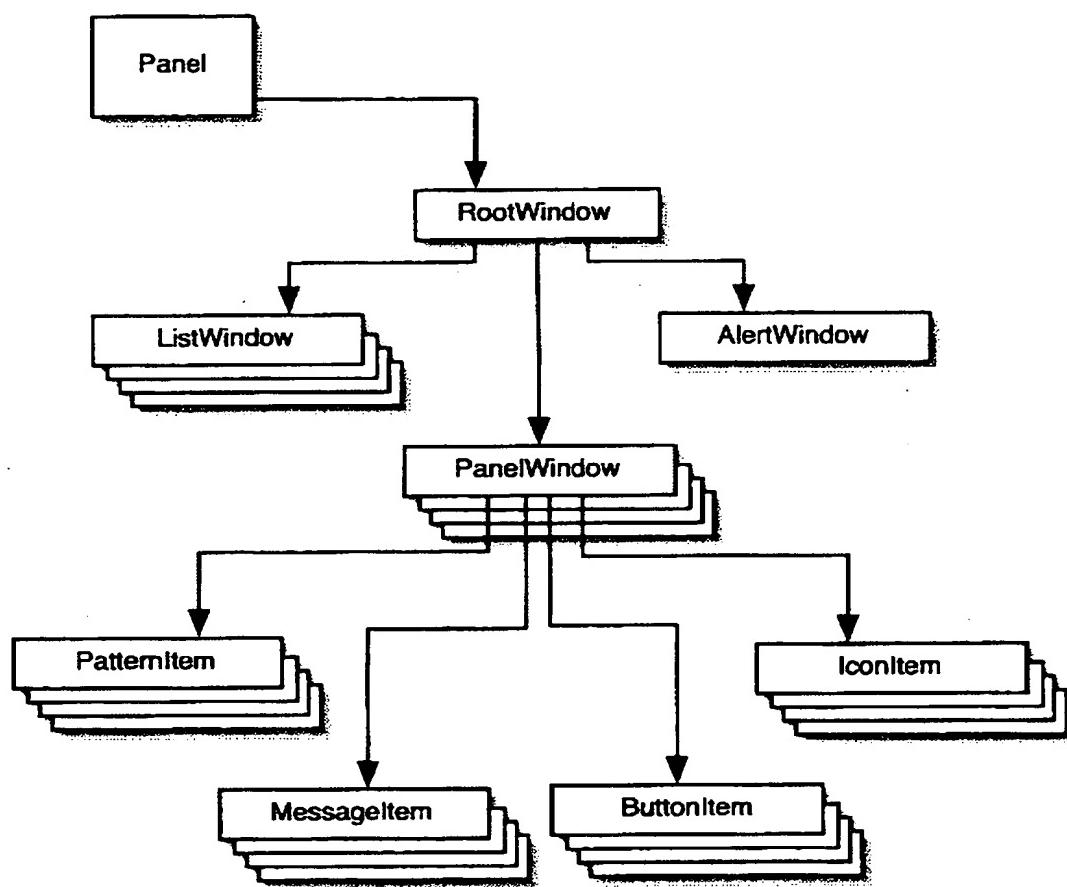
【図 27】

## Java(登録商標)開発環境を説明するための図



【図28】

## 操作パネルクラスの階層構成を示す図



【図 29】

## サンプルプログラムを示す図

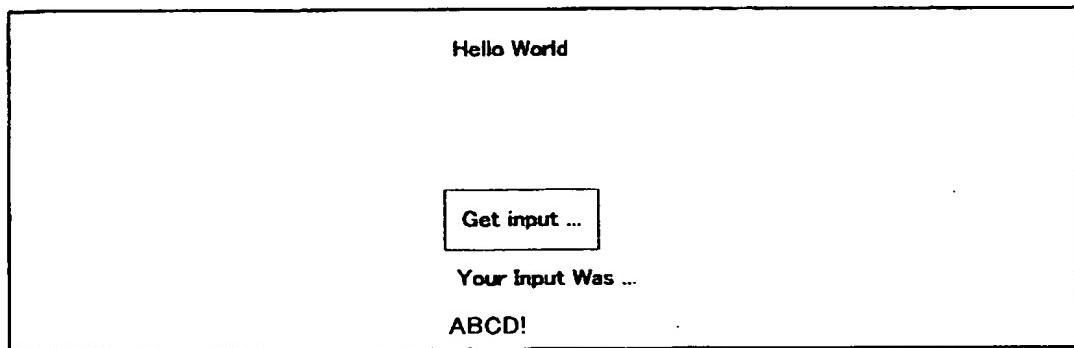
```

import java.lang.*;
import com.hijke.gw.panel.*;
public class UserApp extends GWApp           //←①
{
    public void main()
    {
        Panel panel = new Panel();
        PanelWindow mainwindow =
            new PanelWindow(panel.root());
        ItemString hello = new ItemString("Hello World"); //←③
        MessageItem salutation = new MessageItem(hello);
        salutation.setRect(100, 10, 100, 10);
        mainwindow.addItem(salutation);
        ItemString inputstr =
            new ItemString("Get input..."); //←②
        ButtonItem inputButton = new ButtonItem(); //←④
        inputButton.setButtonShape(ButtonItem.BTN_ZAB);
        inputButton.setRect(100, 100, 100, 100);
        inputButton.setWink(true);
        inputButton.addChangeListener(
            new GWChangeListener()
            {
                public void stateChanged(GMEvent e)
                {
                    String answer =
                        SoftKeyboardWindow.request(
                            "Add your Input",
                            "", 256,
                            ItemString.LANG_ENGLISH_US
                        );
                    inputdisplay.setMessage(
                        new ItemString(answer));
                    inputdisplay.paint();
                }
            });
        mainwindow.addItem(inputButton);
        ItemString label1 =
            new ItemString("Your Input Was..."); //←⑤
        MessageItem label = new MessageItem(label1);
        label.setRect(100, 110, 200, 10);
        mainwindow.addItem(label);
        ItemString outputstr = new ItemString ("");
        MessageItem
        inputdisplay = new MessageItem(outputstr); //←⑥
        inputdisplay.setRect(100, 125, 200, 10);
        mainwindow.addItem(inputdisplay);
    }
}

```

【図 30】

サンプルプログラムの実行結果を示す図



【書類名】 要約書

【要約】

【課題】 オペレーションパネルの操作表示部のカスタマイズを容易に行い、プログラム開発効率を向上させること。

【解決手段】 インターネット 220 に接続されたクライアント端末である PC 200 にオペレーションパネル 210 の操作表示部からの入力に対する実行処理をインタプリタ用言語（シェルスクリプト）で記述したカスタマイズプログラムのプログラミング画面 201 を表示してカスタマイズプログラムを入力させるプログラミングサービス 132 と、入力されたカスタマイズプログラムをステップごとに逐次実行するインタプリタ（シェル） 134 とを備えた。

【選択図】 図 1

特願2002-205378

出願人履歴情報

識別番号 [00006747]

1. 変更年月日 1990年 8月24日

[変更理由] 新規登録

住 所 東京都大田区中馬込1丁目3番6号  
氏 名 株式会社リコー

2. 変更年月日 2002年 5月17日

[変更理由] 住所変更

住 所 東京都大田区中馬込1丁目3番6号  
氏 名 株式会社リコー